

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A205 354



THESIS

DESIGNING FEEDBACK COMPENSATORS BY USING THE ROOT-LOCUS METHOD

by

Levent Korkmaz

December 1988

Thesis Advisor

George J. Thaler

Approved for public release; distribution is unlimited.

DTIC

UNCLASSIFIED

MAR 14 1989



H



Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution Availability of Report Approved for public release; distribution is unlimited.		
2b Declassification Downgrading Schedule			5 Monitoring Organization Report Number(s)		
4 Performing Organization Report Number(s)			7a Name of Monitoring Organization Naval Postgraduate School		
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol (if applicable) 32	7b Address (city, state, and ZIP code) Monterey, CA 93943-5000		
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000		8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number		
8a Name of Funding Sponsoring Organization		8c Address (city, state, and ZIP code)	10 Source of Funding Numbers Program Element No Project No Task No Work Unit Accession No		
11 Title (include security classification) DESIGNING FEEDBACK COMPENSATORS BY USING THE ROOT-LOCUS METHOD					
12 Personal Author(s) Levent Korkmaz					
13a Type of Report Master's Thesis		13b Time Covered From To	14 Date of Report (year, month, day) December 1988		15 Page Count 128
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	Root-Locus, Function Minimization, State Feedback		
19 Abstract (continue on reverse if necessary and identify by block number)					
<p>The purpose of this thesis is to find suitable ways to design feedback compensators for high order systems by using Root-Locus methods.</p> <p>As a starting point we will examine a motor amplidyne system and a position control system that were previously designed using Bode methods. Then we generalize the method and extend it to other systems.</p> <p>The final subject of this thesis is to design feedback compensators as filters by using state feedback coefficients to define zeros of the filter, then we extend this idea to build cascade filters.</p> <p><i>Keywords: Feedback, Root-Locus, State Feedback, Cascade Filters</i></p>					
20 Distribution Availability of Abstract <input checked="" type="checkbox"/> unclassified unlimited <input type="checkbox"/> same as report <input type="checkbox"/> DTIC users			21 Abstract Security Classification Unclassified		
22a Name of Responsible Individual George J. Thaler			22b Telephone (include Area code) (408) 646-2134		22c Office Symbol 621r

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified



Approved for public release; distribution is unlimited.

Designing Feedback Compensators By Using The Root-Locus Method

by

Levent Korkmaz
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1981

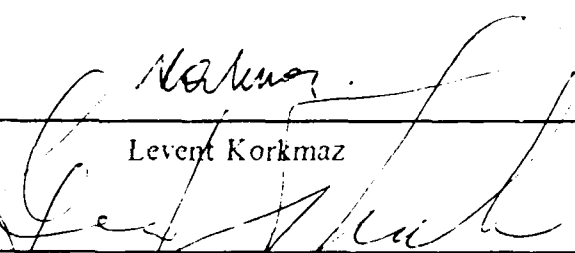
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

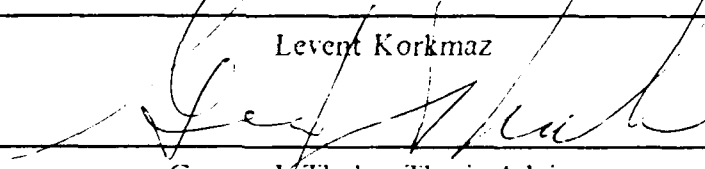
from the

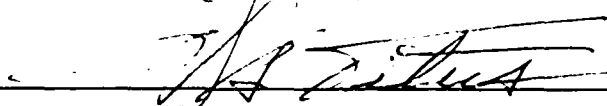
NAVAL POSTGRADUATE SCHOOL
December 1988

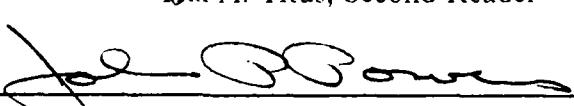
Author:

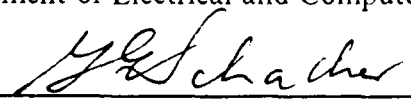

Levent Korkmaz

Approved by:


George J. Thaler, Thesis Advisor


Hal A. Titus, Second Reader


John P. Powers, Chairman,
Department of Electrical and Computer Engineering


Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

The purpose of this thesis is to find suitable ways to design feedback compensators for high order systems by using Root-Locus methods.

As a starting point we will examine a motor amplidyne system and a position control system that were previously designed using Bode methods. Then we generalize the method and extend it to other systems.

The final subject of this thesis is to design feedback compensators as filters by using state feedback coefficients to define zeros of the filter, then we extend this idea to build cascade filters.

Accession For

NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	

By _____

Distribution/ _____

Availability Codes

_____ Avail and/or _____

Dist _____ Total _____

A-1

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PRELIMINARY EXAMPLES	2
A. HIGH GAIN MOTOR AMPLIDYNE SYSTEM	2
B. POSITION CONTROL SYSTEM	16
III. DESIGNING COMPENSATORS	33
A. NON-MINIMUM PHASE SYSTEMS	33
B. FLEXIBLE ROCKET CONTROL SYSTEM	73
IV. DESIGNING FEEDBACK COMPENSATORS USING POLE PLACE- MENT	85
A. ALL COMPONENTS ARE IN THE LEFT HALF PLANE	85
B. NON-MINIMUM PHASE SYSTEMS	91
C. EXTENDING THE IDEA TO CASCADE COMPENSATION	100
V. CONCLUSIONS AND RECOMMENDATIONS	112
A. CONCLUSIONS	112
B. RECOMMENDATIONS	112
APPENDIX A. CONSTRAINT PARAMETERS OF FUNCTION MINIMIZA- TION	113
APPENDIX B. COMPUTER PROGRAM FOR FUNCTION MINIMIZATION	115
LIST OF REFERENCES	117
INITIAL DISTRIBUTION LIST	118

LIST OF FIGURES

Figure 1.	Original Compensated System	2
Figure 2.	Open Loop Bode Plot	3
Figure 3.	Ordinary Manipulation	4
Figure 4.	Comp System Open Loop Bode	5
Figure 5.	Comp System Root-Locus	6
Figure 6.	Comp Sysytem Magnified Root-Locus	7
Figure 7.	Comp System Time response	8
Figure 8.	First Step For Manipulation	9
Figure 9.	Final System For Root-Locus Examination	9
Figure 10.	Uncompensated System Root-Locus	11
Figure 11.	G and 1/H Magnitude Curves	12
Figure 12.	Bode Plot For Loop Transfer Function $GH(s)$	13
Figure 13.	Root-Locus For Loop Transfer Function $GH(s)$	14
Figure 14.	Magnified Root-Locus For Loop Transfer Function $GH(s)$	15
Figure 15.	Original Compensated System	16
Figure 16.	Uncompensated System Open Loop Bode Plot	17
Figure 17.	Ordinary Manipulation	18
Figure 18.	Compensated System Open Loop Bode Plot	19
Figure 19.	Compensated System Root-Locus	20
Figure 20.	Magnified Root-Locus	21
Figure 21.	Compensated System Time Response	22
Figure 22.	First Step For Manipulation	23
Figure 23.	Final System For Root-Locus Examination	23
Figure 24.	Root-Locus For Uncompensated System	26
Figure 25.	G and 1/H Magnitude Curves	27
Figure 26.	Root-Locus For The Loop Transfer Function $GH(s)$	28
Figure 27.	Magnified Root-Locus For The Loop Transfer Function $GH(s)$	29
Figure 28.	The Same System With Two Zeros at Origin.	30
Figure 29.	Two Zeros Pole at $s = -50$	31
Figure 30.	Two Zeros Pole at $s = -75$	32
Figure 31.	Original System	33

Figure 32. Uncompensated System Open Loop Bode	34
Figure 33. Root-Locus For Uncompensated System	35
Figure 34. Magnified Root-Locus For Uncompensated System	36
Figure 35. Loop Transfer Function Form	37
Figure 36. Root-Locus For Loop Transfer Function	38
Figure 37. Magnified Root-Locus For Loop Transfer Function	39
Figure 38. Compensated System	40
Figure 39. Resulting Equivalent G	40
Figure 40. Root-Locus For Final System	41
Figure 41. Magnified Root-Locus For Final System	42
Figure 42. Bode Plot Without Outside Gain	43
Figure 43. Time Response Without Outside Gain	44
Figure 44. Final Compensated System	45
Figure 45. Bode Plot For Final System Loop Gain = 100	46
Figure 46. Final System Loop Gain = 100	47
Figure 47. Loop Transfer Function Form	48
Figure 48. Root-Locus For Loop Transfer Function	49
Figure 49. Magnified Root-Locus For Loop Transfer Function	50
Figure 50. Compensated System	51
Figure 51. Resulting Equivalent G	51
Figure 52. Root-Locus For Final System	52
Figure 53. Magnified Root-Locus For Final System	53
Figure 54. Bode Plot Without Outside Gain	54
Figure 55. Time Response Without Outside Gain	55
Figure 56. Final Compensated System	56
Figure 57. Bode Plot For Final System Loop Gain = 100	57
Figure 58. Final System Loop Gain = 100	58
Figure 59. Loop Transfer Function Form	59
Figure 60. Root-Locus For Loop Transfer Function	60
Figure 61. Magnified Root-Locus For Loop Transfer Function	61
Figure 62. Compensated System	62
Figure 63. Resulting Equivalent G	63
Figure 64. Bode Plot Without Outside Gain	64
Figure 65. Time Response Without Outside Gain	65
Figure 66. Final Compensated System	66

Figure 67. Bode Plot For Final System Loop Gain = 100	67
Figure 68. Final System Loop Gain = 100	68
Figure 69. Final System Loop Gain = 500	70
Figure 70. Final System Loop Gain = 1000	71
Figure 71. Original Uncompensated System	72
Figure 72. Combined Uncompensated System	73
Figure 73. Root-Locus For Uncompensated System	74
Figure 74. Illustration of General Configuration	75
Figure 75. Root-Locus for General Type Compensated System	76
Figure 76. First Resulting Compensator of Function Minimization	77
Figure 77. Root-Locus for First FM Result	78
Figure 78. Time Response for First FM Result	79
Figure 79. Second compensator Resulting from Function Minimization.	80
Figure 80. Root-Locus for Second FM Result	81
Figure 81. Time Response for Second FM Result	82
Figure 82. Block Diagram of Total Compensated System	83
Figure 83. State Feedback Compensated System.	85
Figure 84. Root-Locus For State Feedback Loop Transfer Function.	86
Figure 85. The Form of The System With Unity Feedback.	87
Figure 86. System With Extra Poles.	88
Figure 87. Root-Locus For Filter Feedback Loop Transfer Function.	89
Figure 88. Final Compensated System	90
Figure 89. Time Response for Only State Feedback Compensated System	91
Figure 90. Time Response for Filter Compensated System	92
Figure 91. State Feedback to Compensate The System	94
Figure 92. Root-Locus For State Feedback Loop Transfer Function.	95
Figure 93. The Form of The System With Unity Feedback.	96
Figure 94. System With Extra Poles	96
Figure 95. Root-Locus For Filter Feedback Loop Transfer Function.	98
Figure 96. Final Compensated System	99
Figure 97. Time Response For Only State Feedback Compensated System	100
Figure 98. Time Response For Filter Compensated System	101
Figure 99. Cascade Compensated System	102
Figure 100. Time Response for Cascade Compensator	103
Figure 101. Cascade Compensated Second System	104

Figure 102. Time Response For Cascade Compensator (Second example)	105
Figure 103. ALCON Simulation of First Problem	107
Figure 104. ALCON Simulation of Second Problem.	108
Figure 105. Second DSL Simulation of First Problem	109
Figure 106. Second DSL Simulation of Second Problem	110

ACKNOWLEDGEMENTS

A significant debt of gratitude is owed to Dr. George J. Thaler, for the many hours of assistance and guidance he has extended, from the author's first course in control theory through more advanced courses, and especially for his help in the preparation of this thesis.

Also I would like to express my sincere appreciation to Dr. Hal A. Titus of the Department of Electrical and Computer Engineering of the Naval Postgraduate School, my second reader.

To my wife, Huray Korkmaz and my family, for their encouragement and patience, I am deeply grateful.

Finally, I wish to express my appreciation to the Turkish Navy Authority for the opportunity to study in the Naval Postgraduate School.

I. INTRODUCTION

Modern control systems may be compensated by placing a suitable filter in either the forward (cascade) path or in a feedback path.

Mainly feedback systems have the advantage that the output follows the command more accurately, so the steady state error is less, the bandwidth is greater, and the speed of response is faster. Also we can say that the effect of external disturbances and internal changes in the parameters or structure of the system is significantly reduced.

The conventional and more common way to design a Feedback compensator to meet a given set of specifications is the Bode plot method, by determining a suitable gain cross-over frequency and $1/H$ curve.

In this thesis we will try to find some rules to do the same design by using Root-Locus methods. To achieve this, first we will examine several designs then try to put the general ideas in a set of rules.

Simulation studies employed Dynamic Simulation Language (DSL) and Ewald package in the IBM 3033, 4381 main frame.

II. PRELIMINARY EXAMPLES

A. HIGH GAIN MOTOR AMPLIDYNE SYSTEM

To study the ideas for feedback compensation first we choose an industrial design that exists and is available for comparison. Figure 1 shows the whole compensated system, where

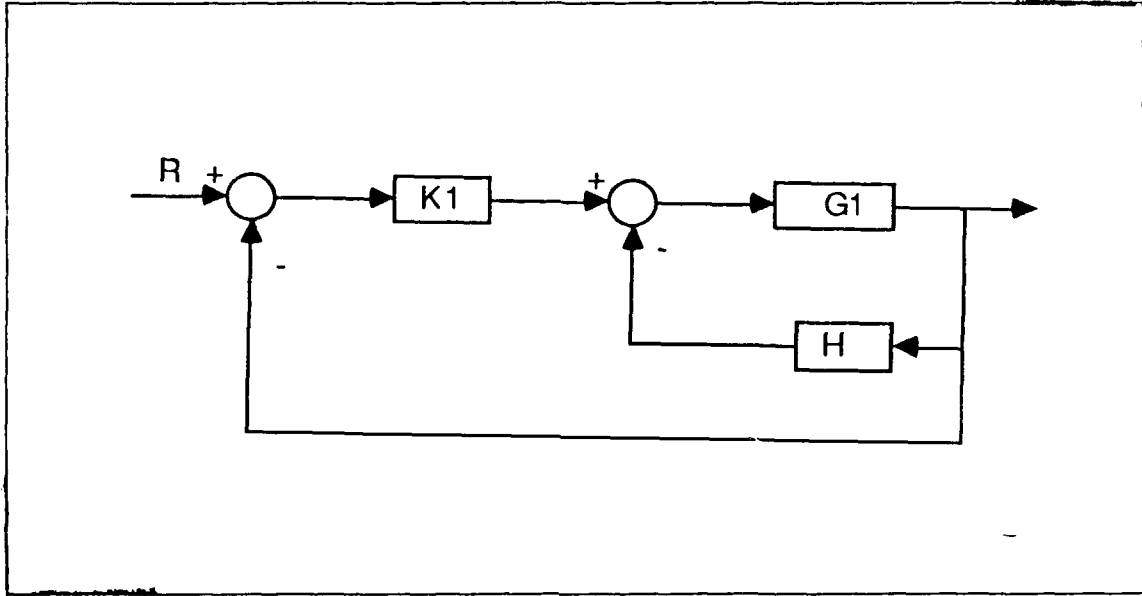


Figure 1. Original Compensated System

$$K_1 = 14.8 \quad (1)$$

$$G_1(s) = \frac{1209.8}{s(s + 57.96)(s + 0.12 \mp j9.32)} \quad (2)$$

$$H(s) = \frac{0.137s^3(s + 62.5)(s + 71.65)}{(s + 1)(s + 5.26)(s + 625)} \quad (3)$$

Figure 2 shows the open loop Bode plot for the uncompensated system. The uncompensated system is badly unstable.

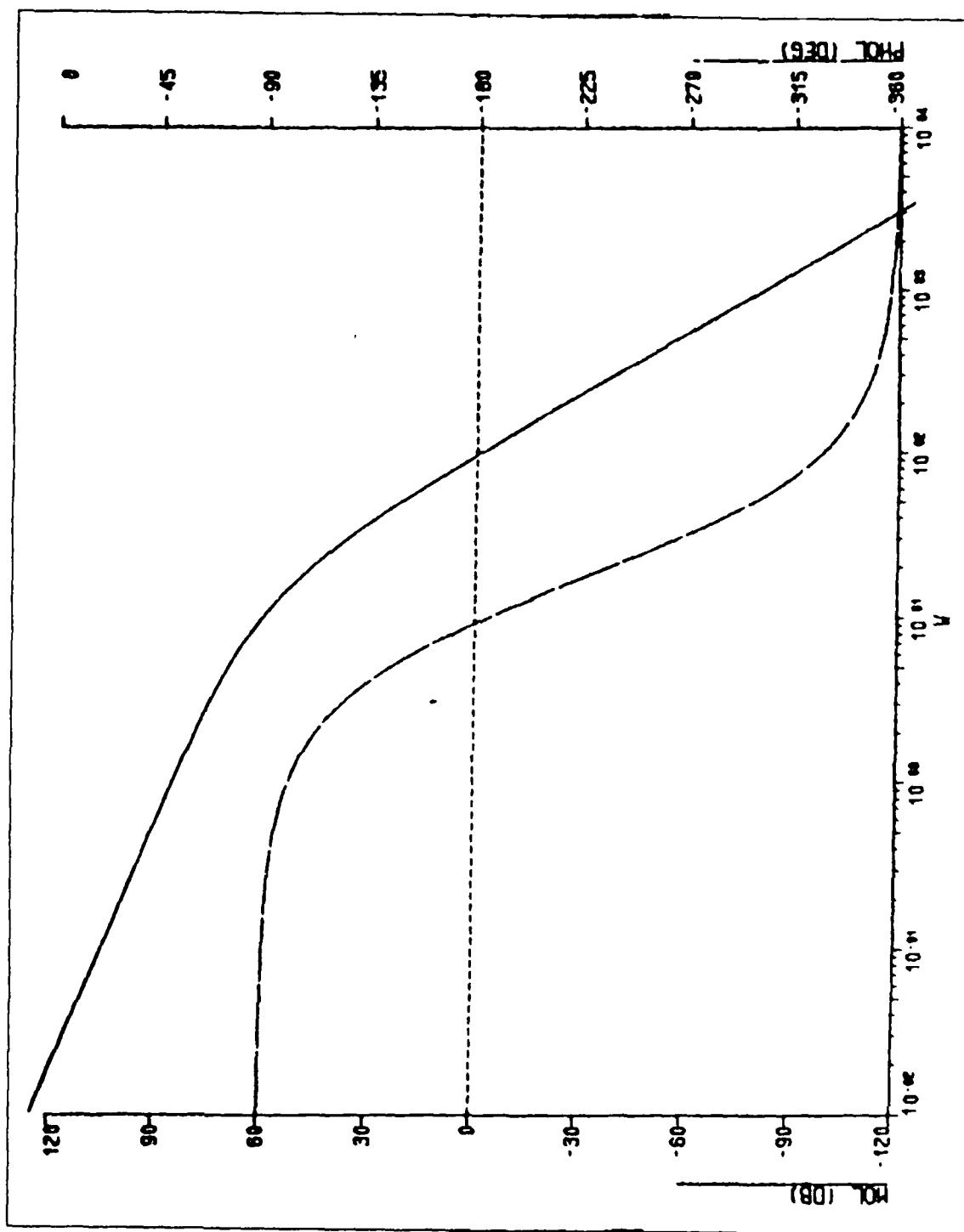


Figure 2. Open Loop Bode Plot

We may use two different kinds of block diagram manipulation to analyze this system. First is the ordinary manipulation shown at Figure 3, where

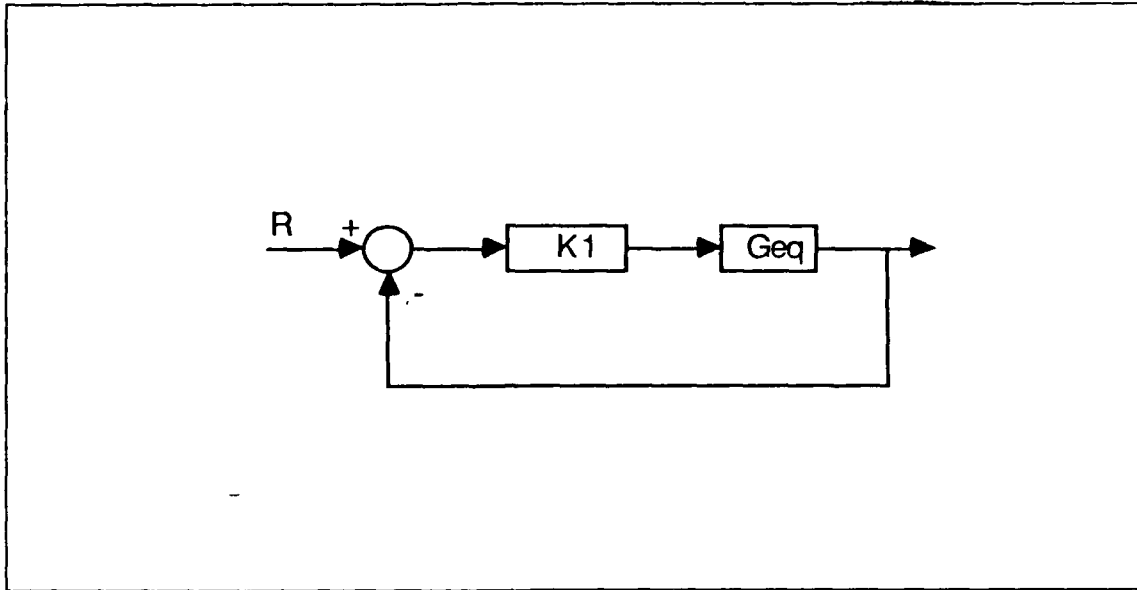


Figure 3. Ordinary Manipulation

$$K_1 = 14.8 \quad (4)$$

$$G_{eq} = \frac{6096553.14(s+1)(s+5.26)(s+625)}{s(s+0.003\mp j0.066)(s+59.35)(s+79.2)(s+275.45\mp j845.8)} \quad (5)$$

Analysis of this transfer function provides the open loop Bode plot, Root-Locus, magnified Root-Locus and time response of the system which are given by Figures 4 - 7. Since there is no pole in the right half plane, we can get phase margin and gain margin from these plots.

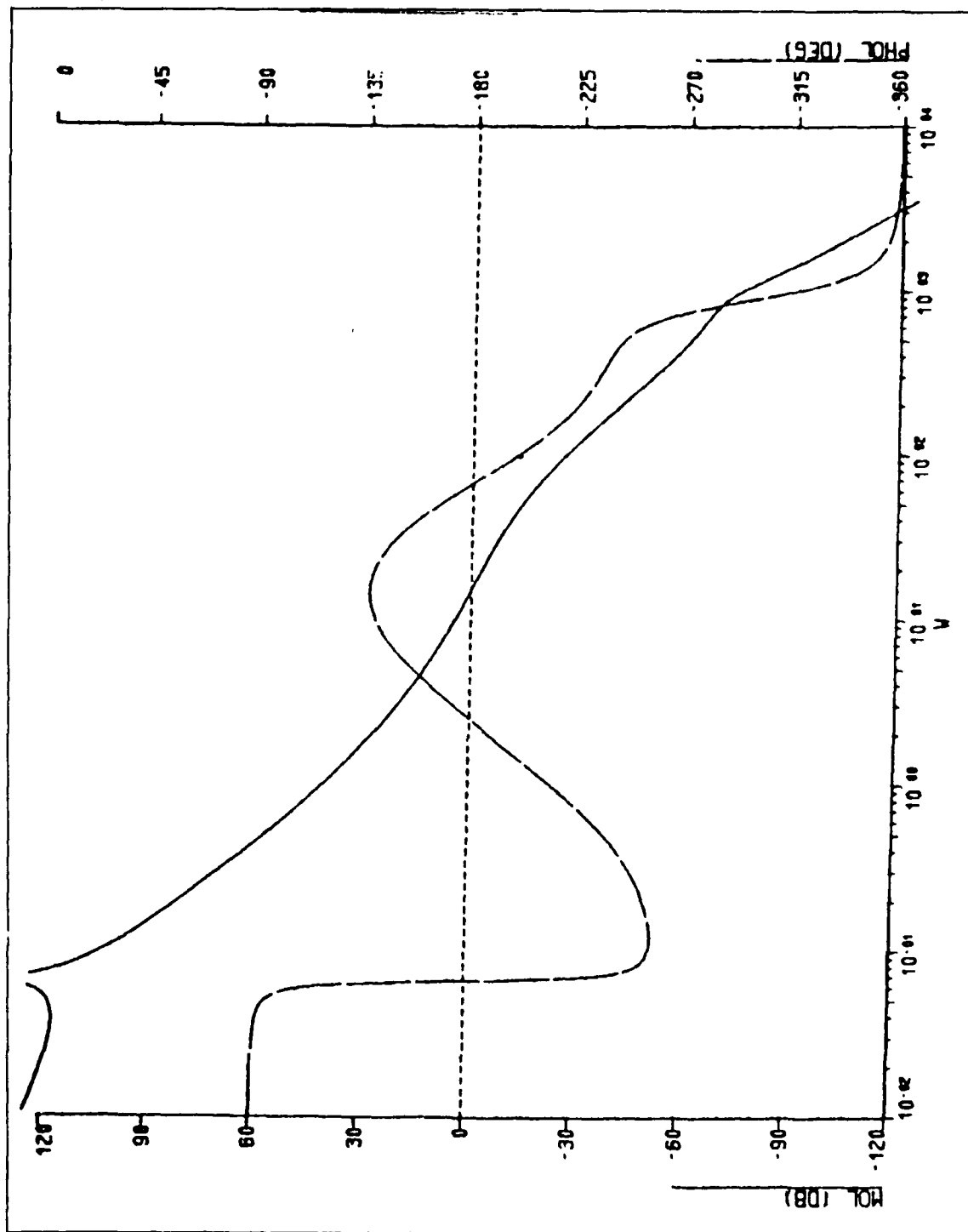


Figure 4. Comp System Open Loop Bode

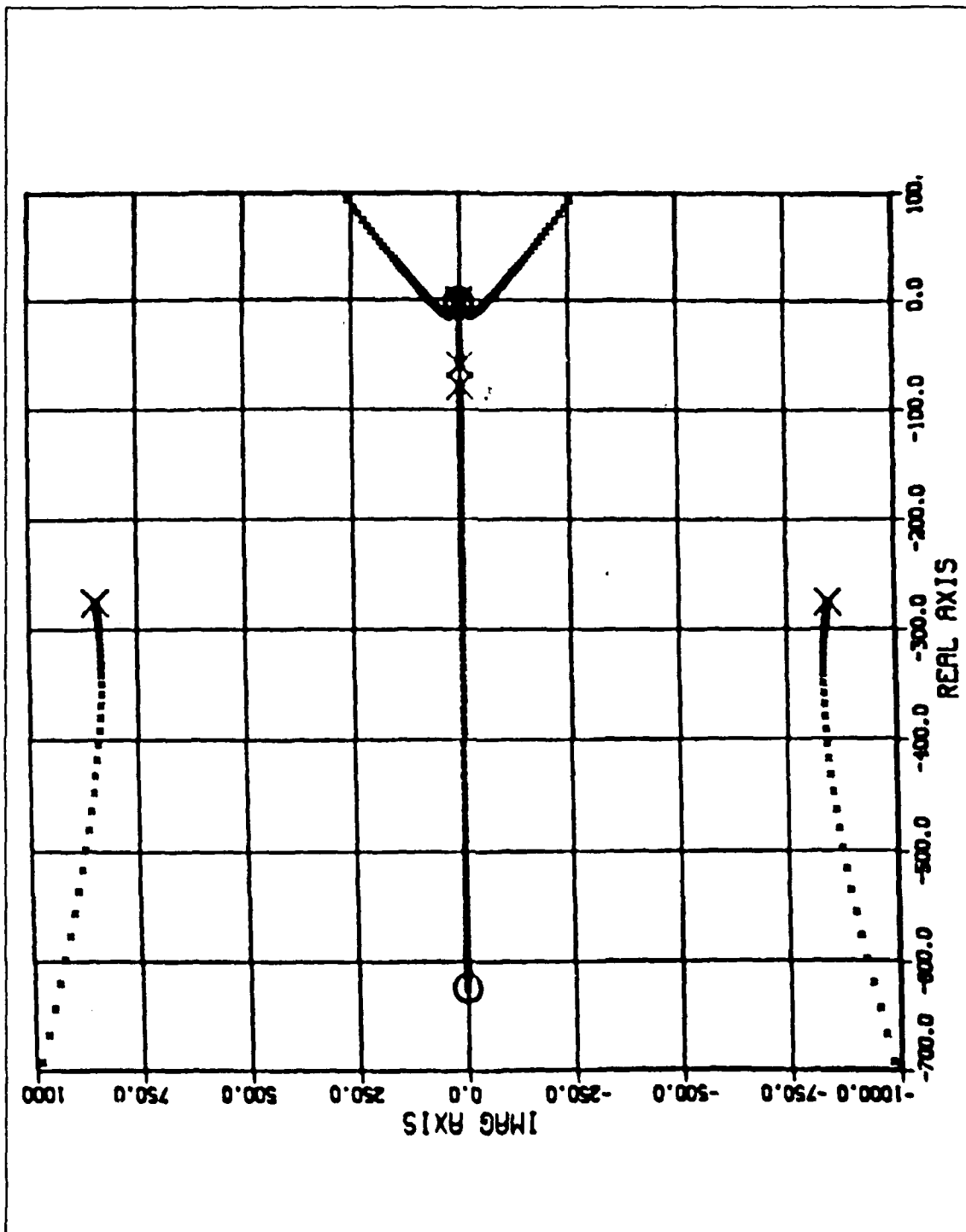


Figure 5. Comp System Root-Locus

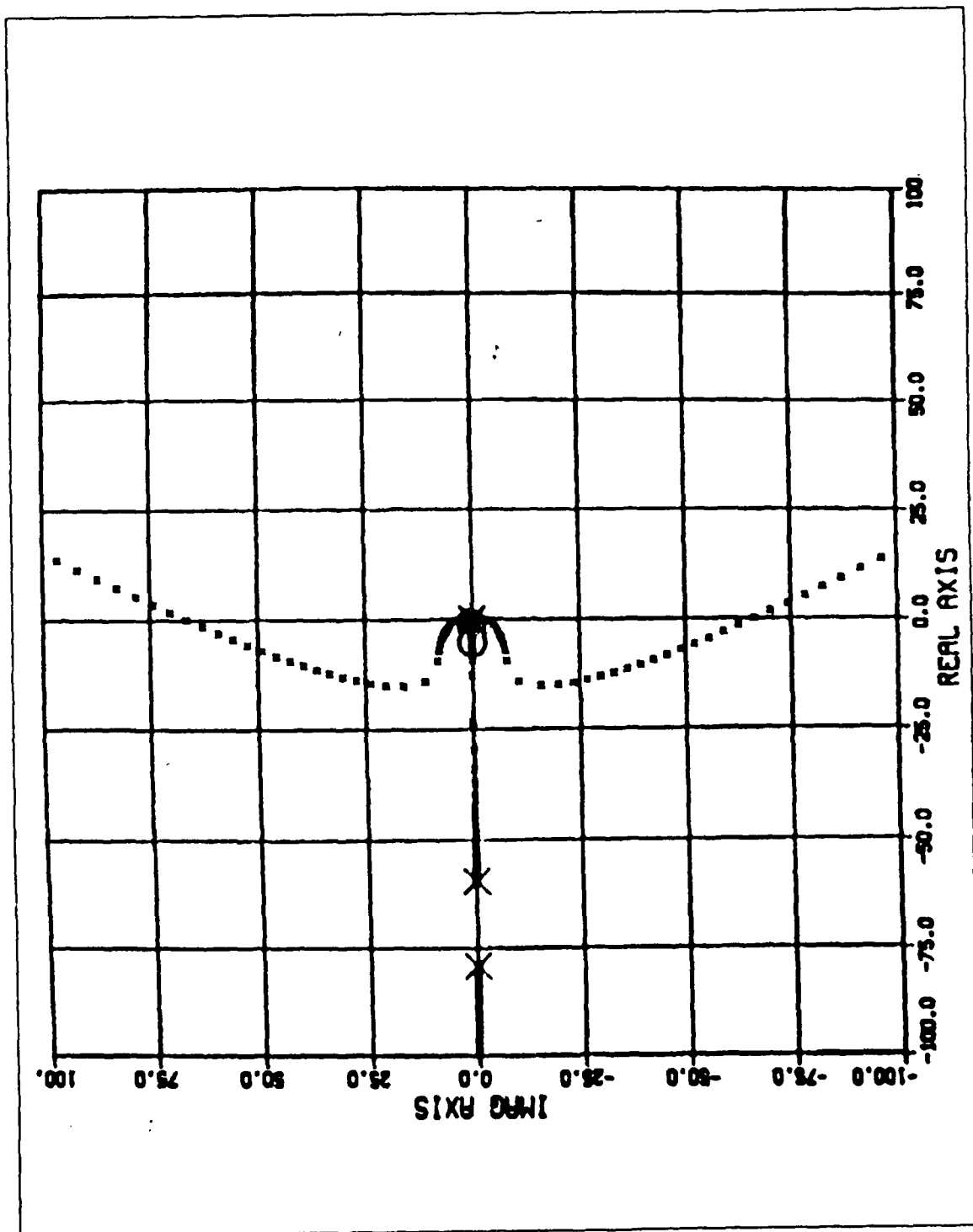


Figure 6. Comp Sysytem Magnified Root-Locus

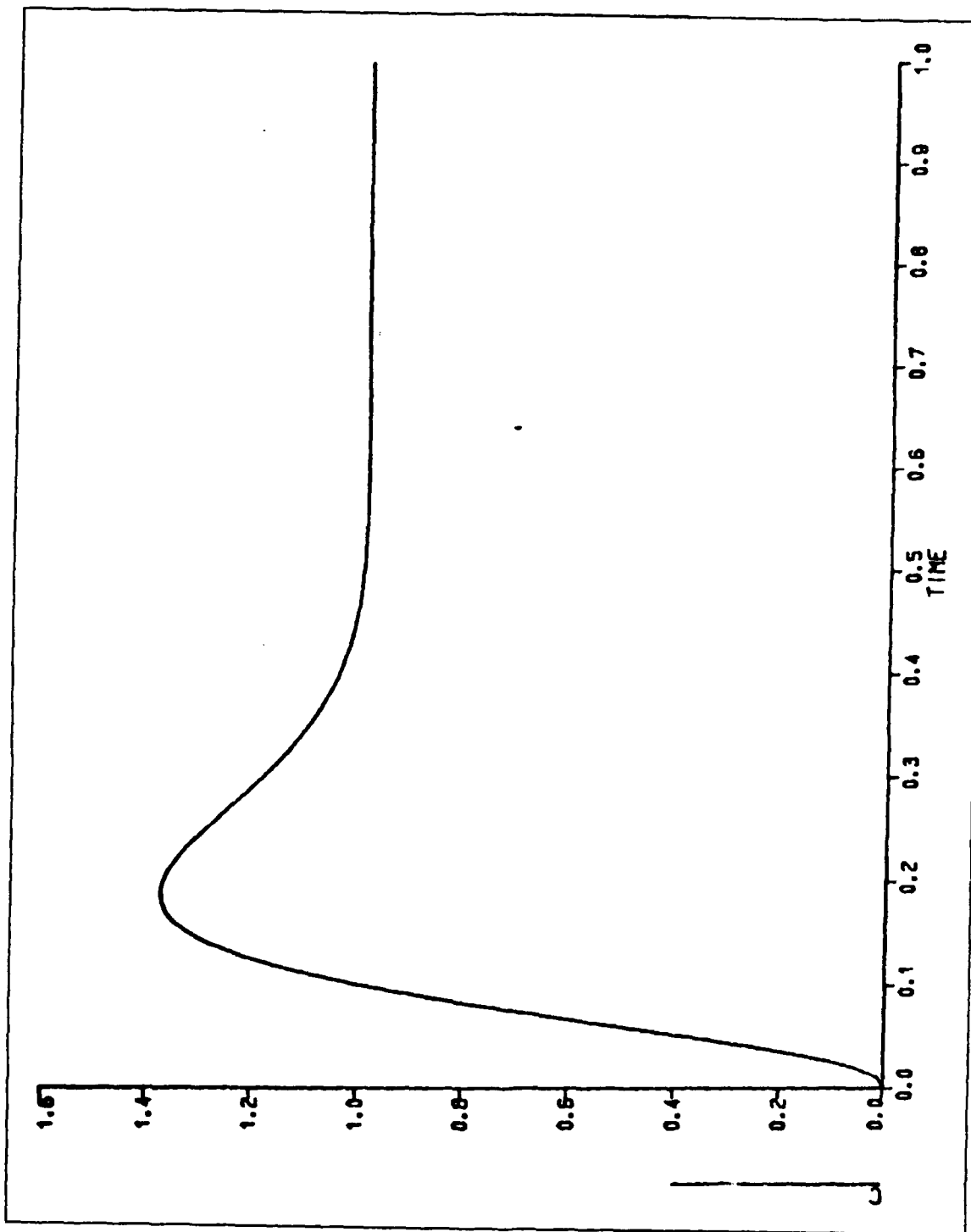


Figure 7. Comp System Time response

The second type of manipulation is done to get more information using the Root-Locus method. The first step for this is shown at Figure 8. Then with the appropriate block diagram reduction the system becomes as in Figure 9, where

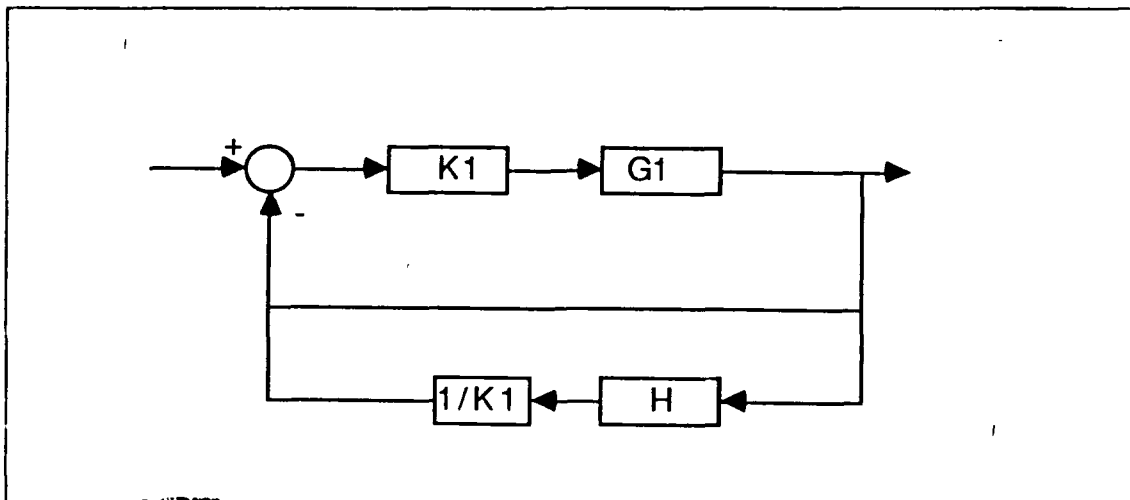


Figure 8. First Step For Manipulation

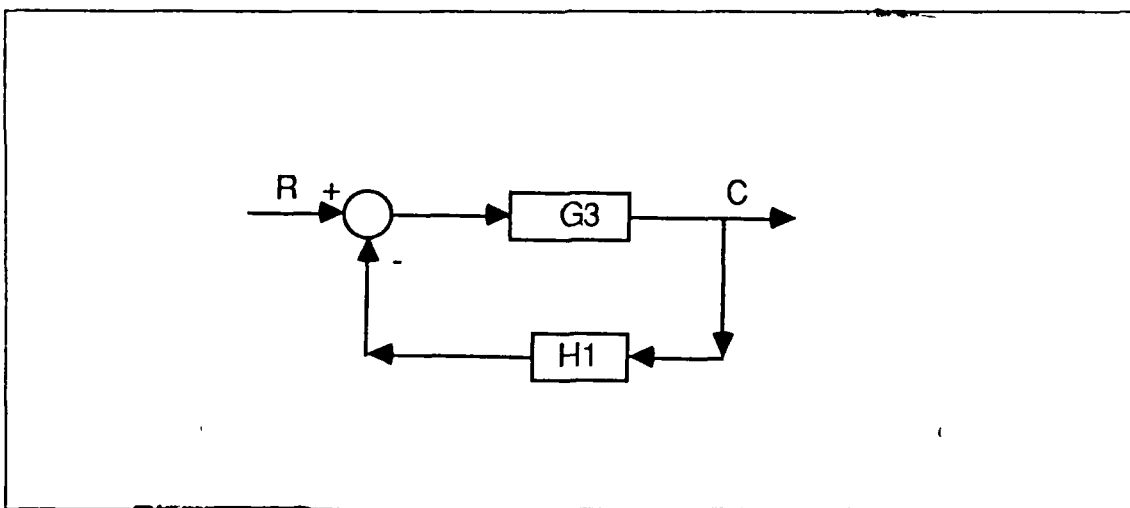


Figure 9. Final System For Root-Locus Examination

$$G3(s) = \frac{17905}{(s + 579.97)(s + 56.14)(s - 26.98 \mp j45.19)} \quad (6)$$

$$H1(s) = \frac{0.00926s^3(s + 62.16)(s + 71.92)}{(s + 1)(s + 5.24)(s + 625.33)} \quad (7)$$

By using the loop transfer function $G11(s)$ to draw the Root-Locus we may examine the system for root movements. Since the $G(s)$ has two poles in the right half plane, the Bode plot cannot be interpreted from the point of view of gain margin or phase margin, but it shows the stability of the system.

The root locus for the uncompensated system is shown at Figure 10.

From this point we can proceed to explore the design procedure of the designer. First of all, since this design was done using Bode design methods, it will be helpful to get the $G1$ and $1/H1$ magnitude curve which is shown at Figure 11 for the original system at Figure 1.

As we may see from the block diagram there are zeros at the origin. For this system we need at least two zeros to keep the system type number unchanged. Also the specifications show that we want to have a very high error coefficient for steady state accuracy and a very narrow bandwidth. Because of these characteristics, there is a dipole near the origin. It is possible to see this on the Bode plot of Figure 11.

The designer could have put the third zero very close to origin but in that case building the system might difficult.

Since other poles are far away compared with these zeros and dipole, these components will decide the system behaviour.

The Bode plot, Root-Locus and magnified Root-Locus around the origin for the loop transfer function $G11(s)$ are given by Figures 12 - 14. Since we have roots in the right half plane, the Bode plot cannot be interpreted for phase and gain margin values.

As a result of this problem we may say that for high error coefficient and narrow bandwidth systems we may need zeros at the origin and a dipole very close to the origin.

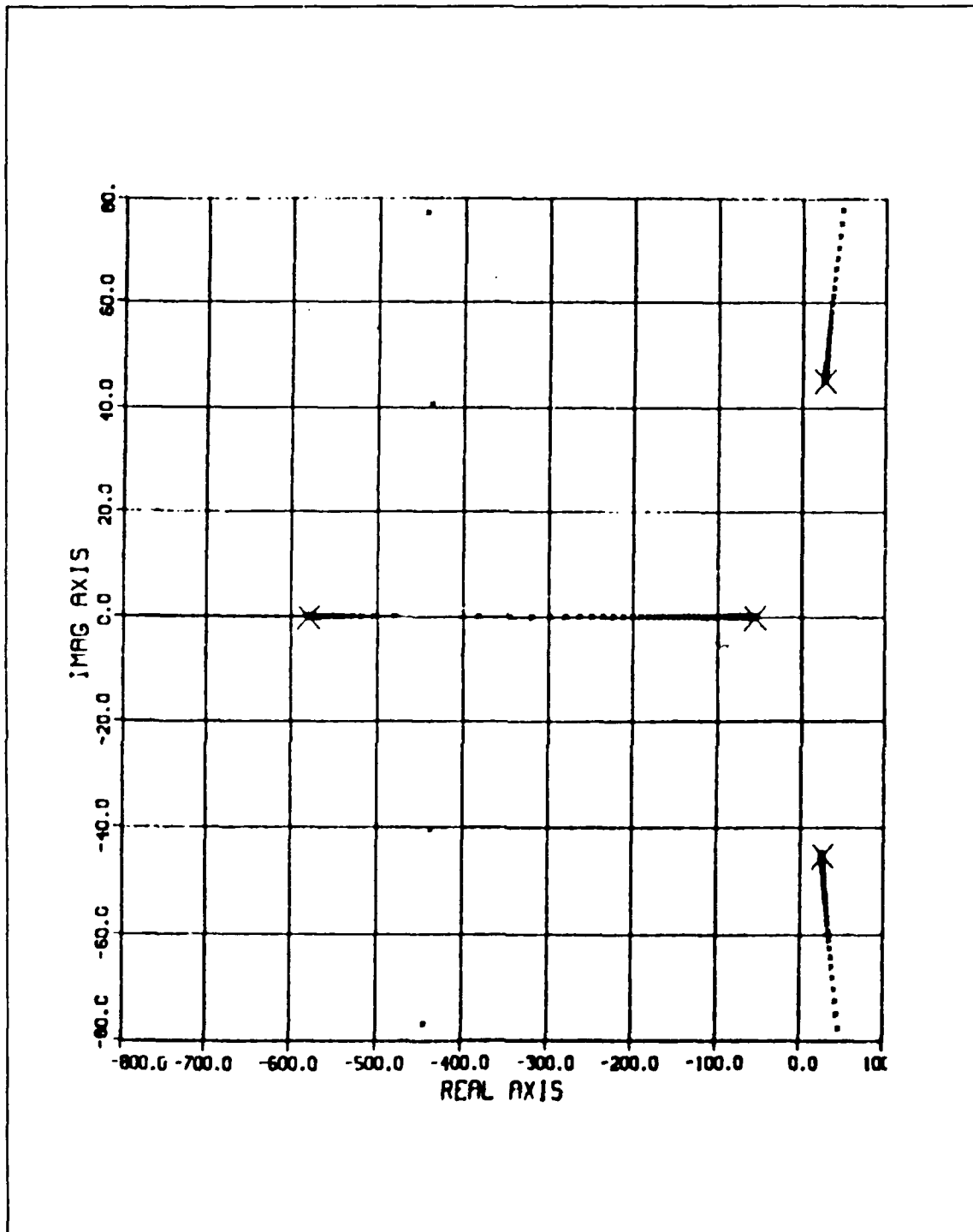


Figure 10. Uncompensated System Root-Locus

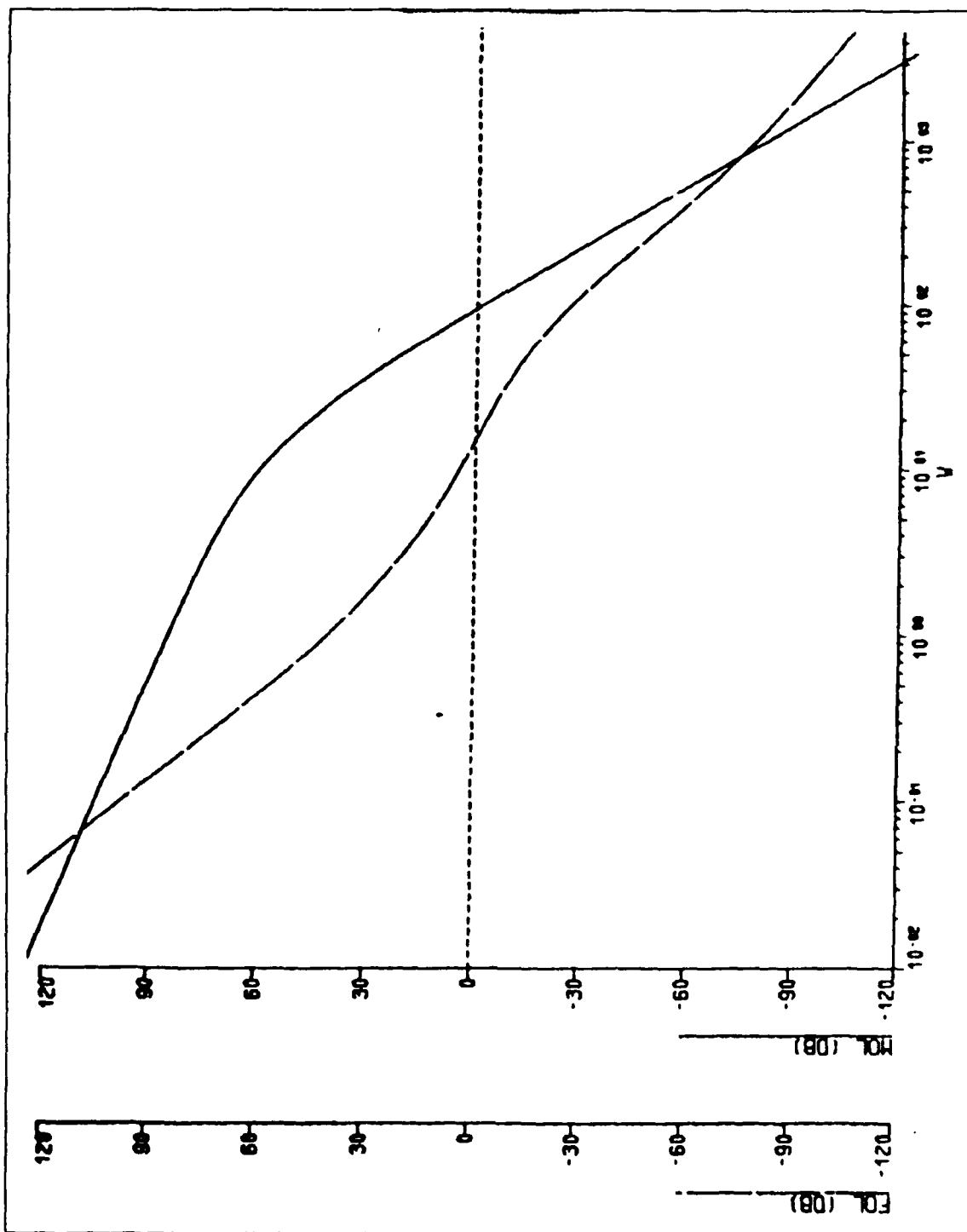


Figure 11. G and $1/H$ Magnitude Curves

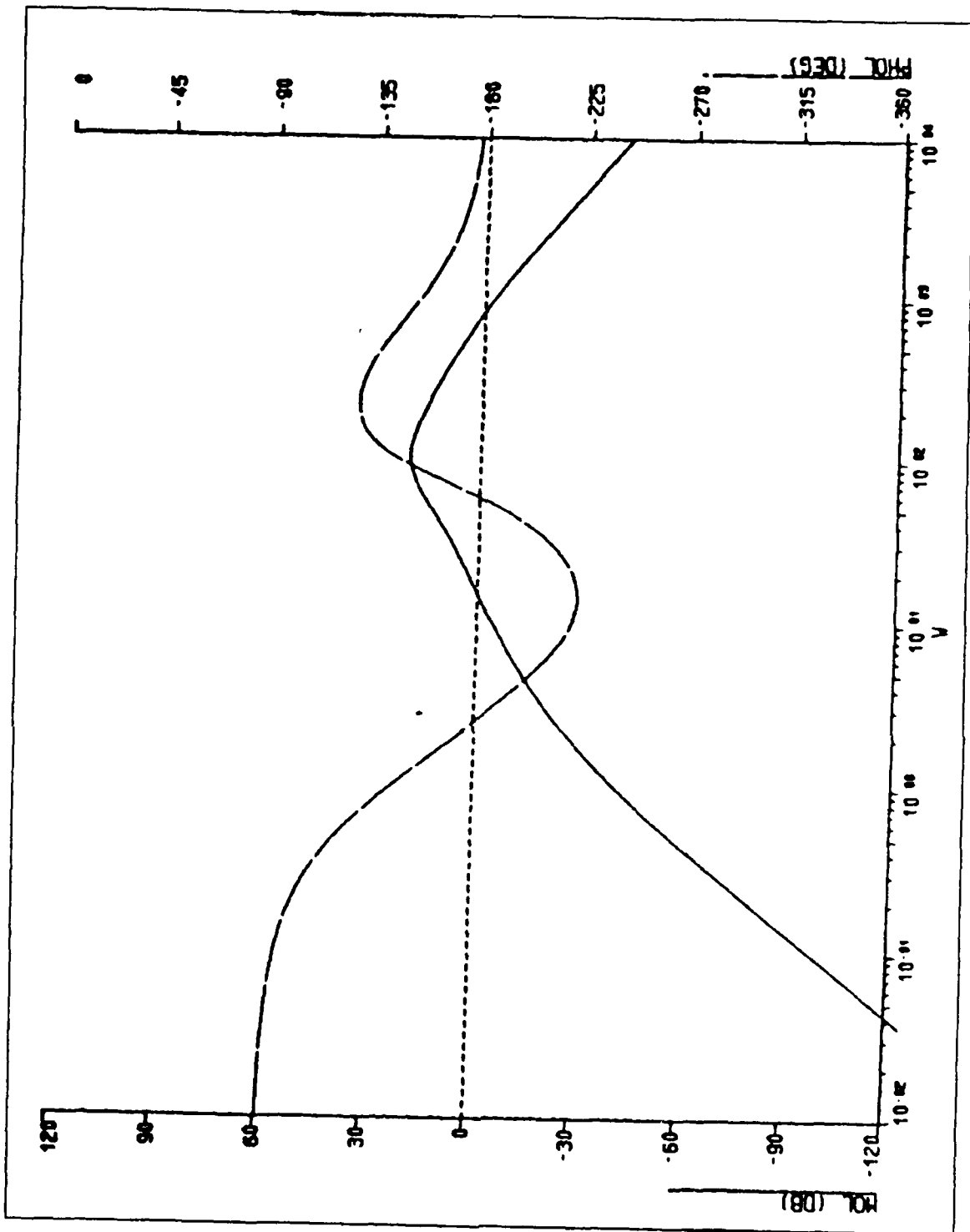


Figure 12. Bode Plot For Loop Transfer Function $GH(s)$

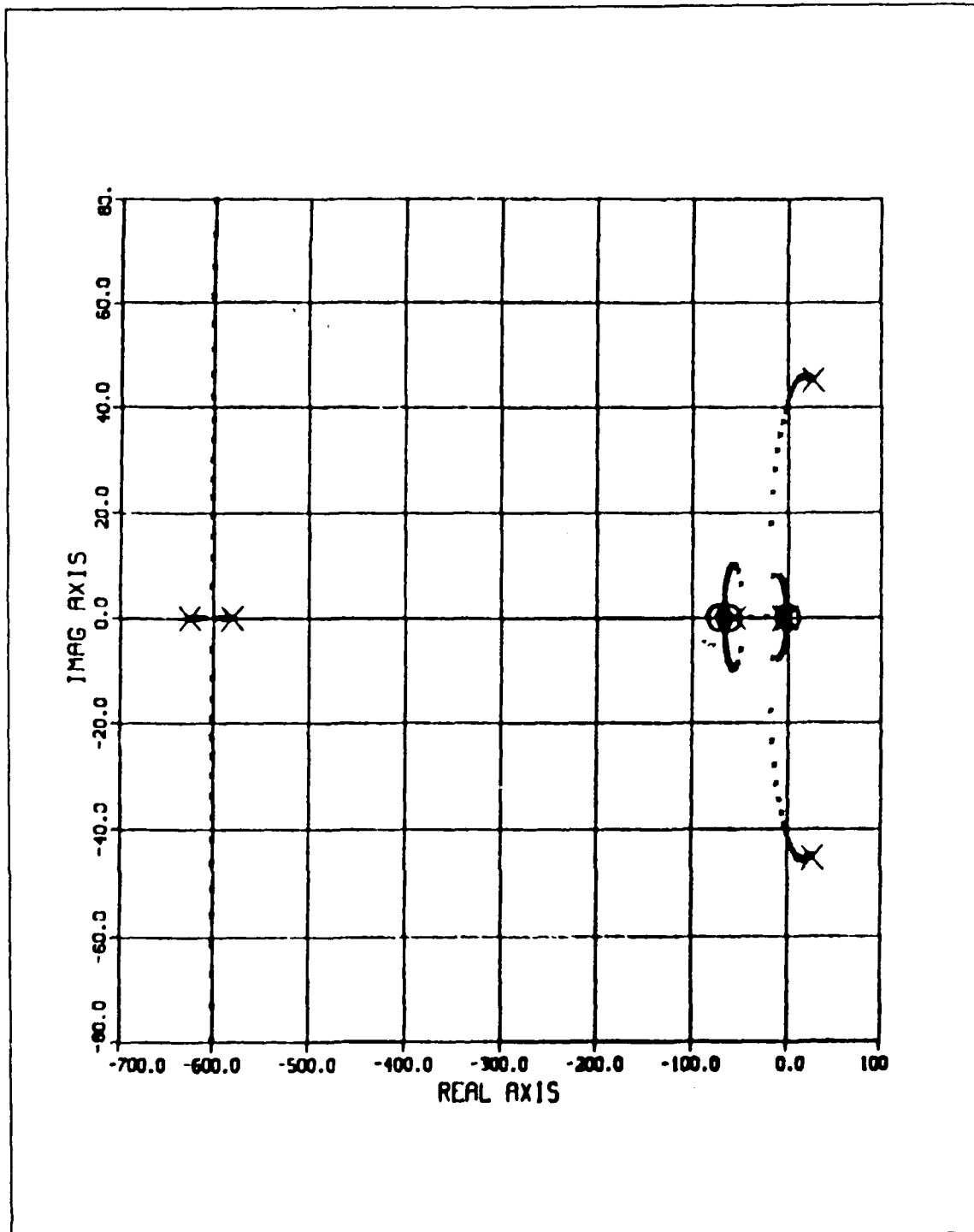


Figure 13. Root-Locus For Loop Transfer Function $GH(s)$

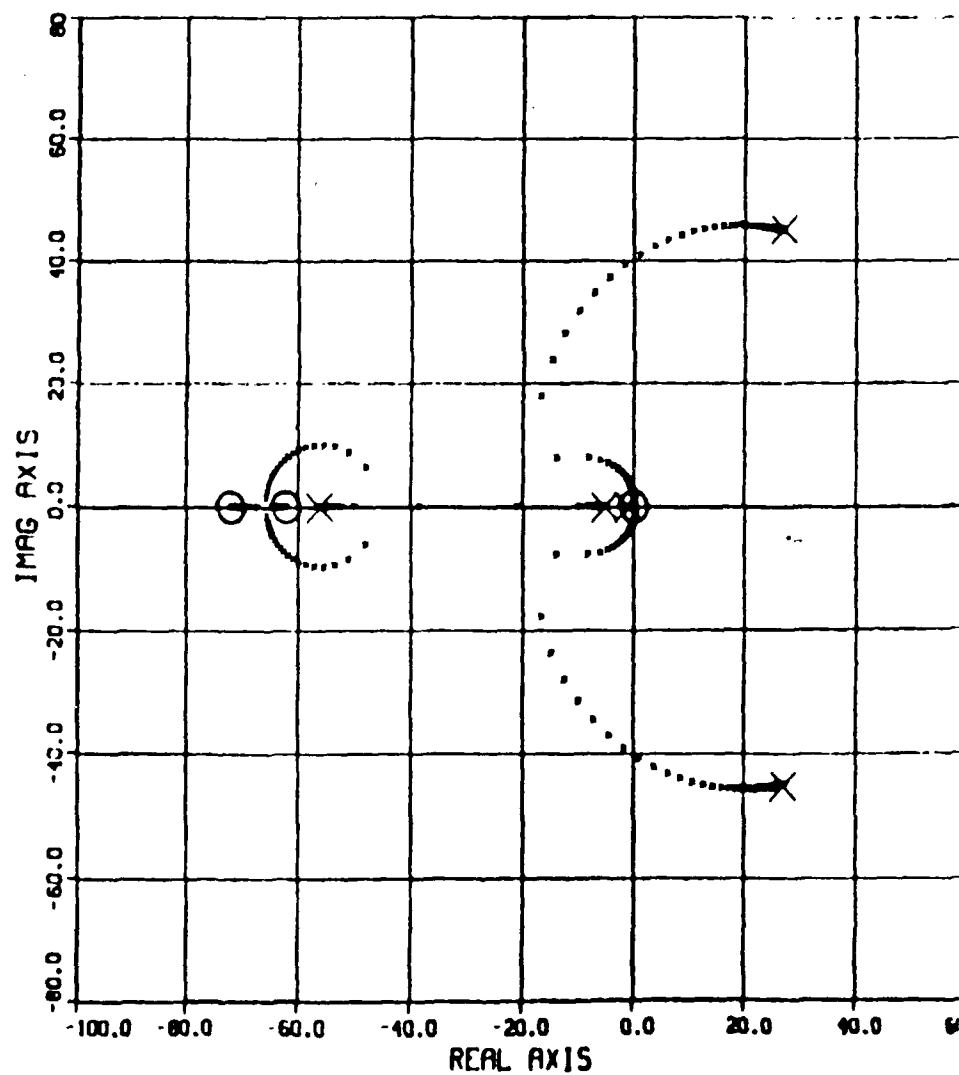


Figure 14. Magnified Root-Locus For Loop Transfer Function $GH(s)$

B. POSITION CONTROL SYSTEM

As a second example we chose a position control system having an amplidyne and a DC armature controlled motor as its power element. Figure 15 shows the original compensated system, where

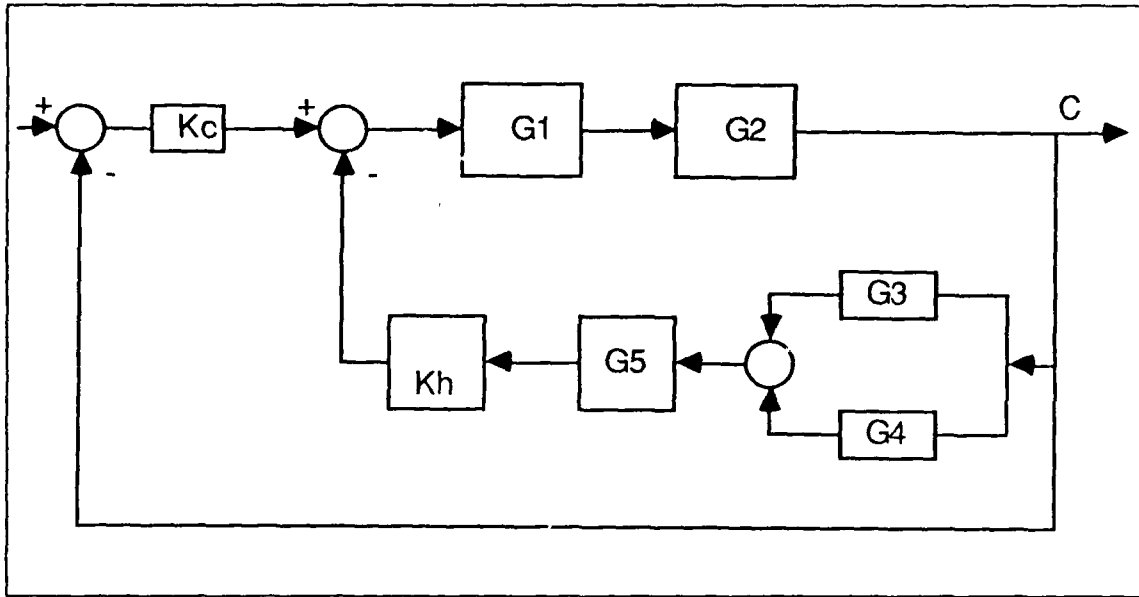


Figure 15. Original Compensated System

$$K_c = 0.233 \quad (8)$$

$$K_h = 0.061 \quad (9)$$

$$G_1 = \frac{12000}{0.078s + 1} \quad (10)$$

$$G_2 = \frac{1}{1.88s(0.0402s + 1)} \quad (11)$$

$$G_3 = 0.333s \quad (12)$$

$$G_4 = 0.0046s^2 \quad (13)$$

$$G_5 = \frac{s^2}{s^2 + 6.5s + 10} \quad (14)$$

Figure 16 shows the open loop Bode plot for the uncompensated system. The uncompensated system is badly unstable.

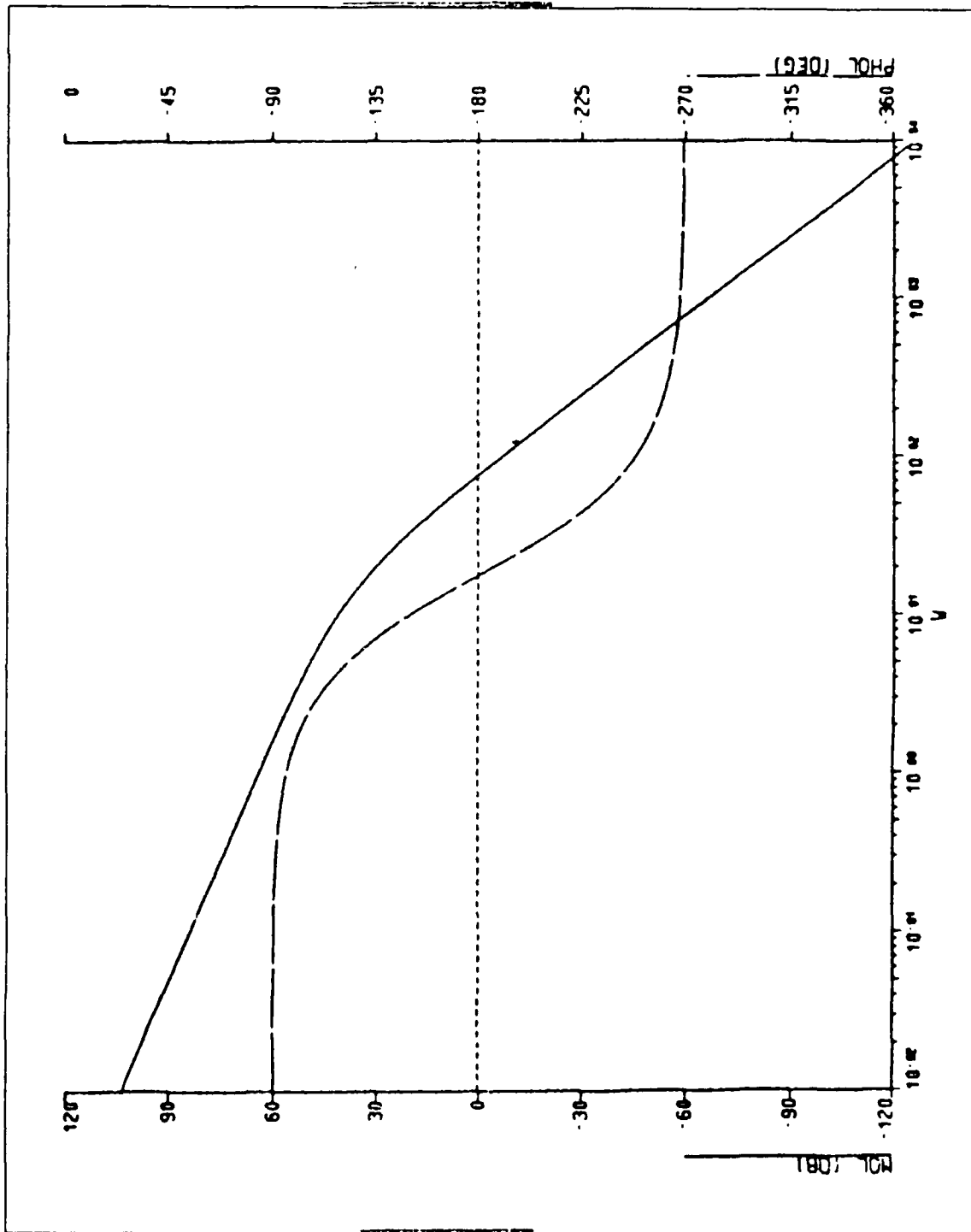


Figure 16. Uncompensated System Open Loop Bode Plot

Again we may use two different kinds of block diagram manipulation to analyze this system.

First is the ordinary manipulation shown at Figure 17, where

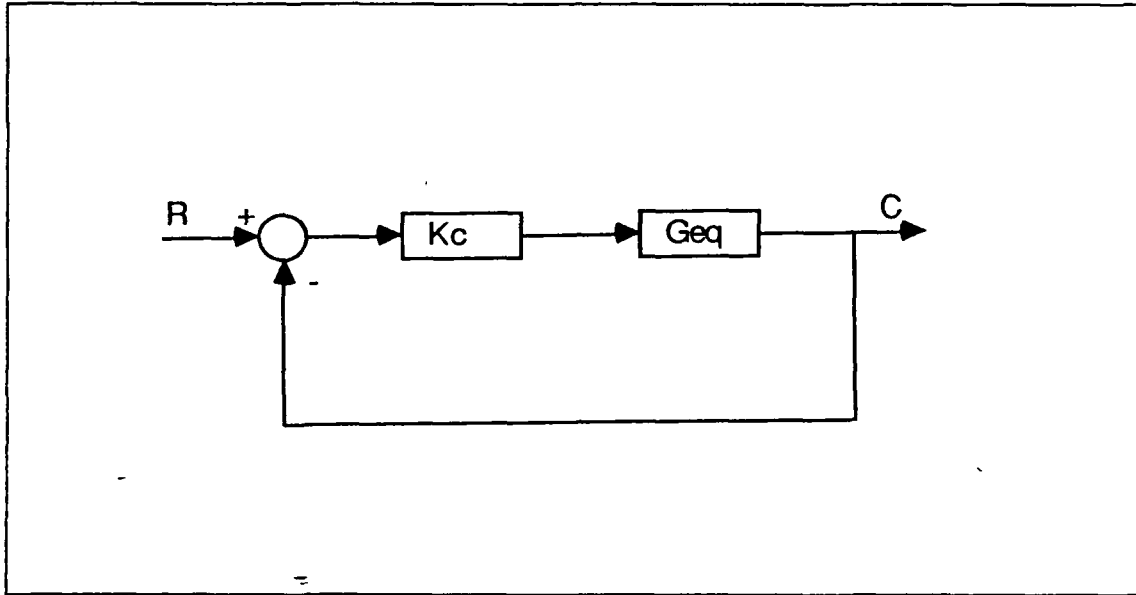


Figure 17. Ordinary Manipulation

$$K_c = 0.233$$

$$G_{eq} = \frac{12000(s^2 + 6.5s + 10)}{0.0059s^5 + 3.62855s^4 + 247.1445s^3 + 14.45s^2 + 18.8s} \quad (16)$$

Analysis of this transfer function provides the open loop Bode plot, Root-Locus, magnified Root-Locus, and time response of the system which are given by Figures 18 - 21

Since there is no root in the right half plane, we can get phase and gain margins from this plot.

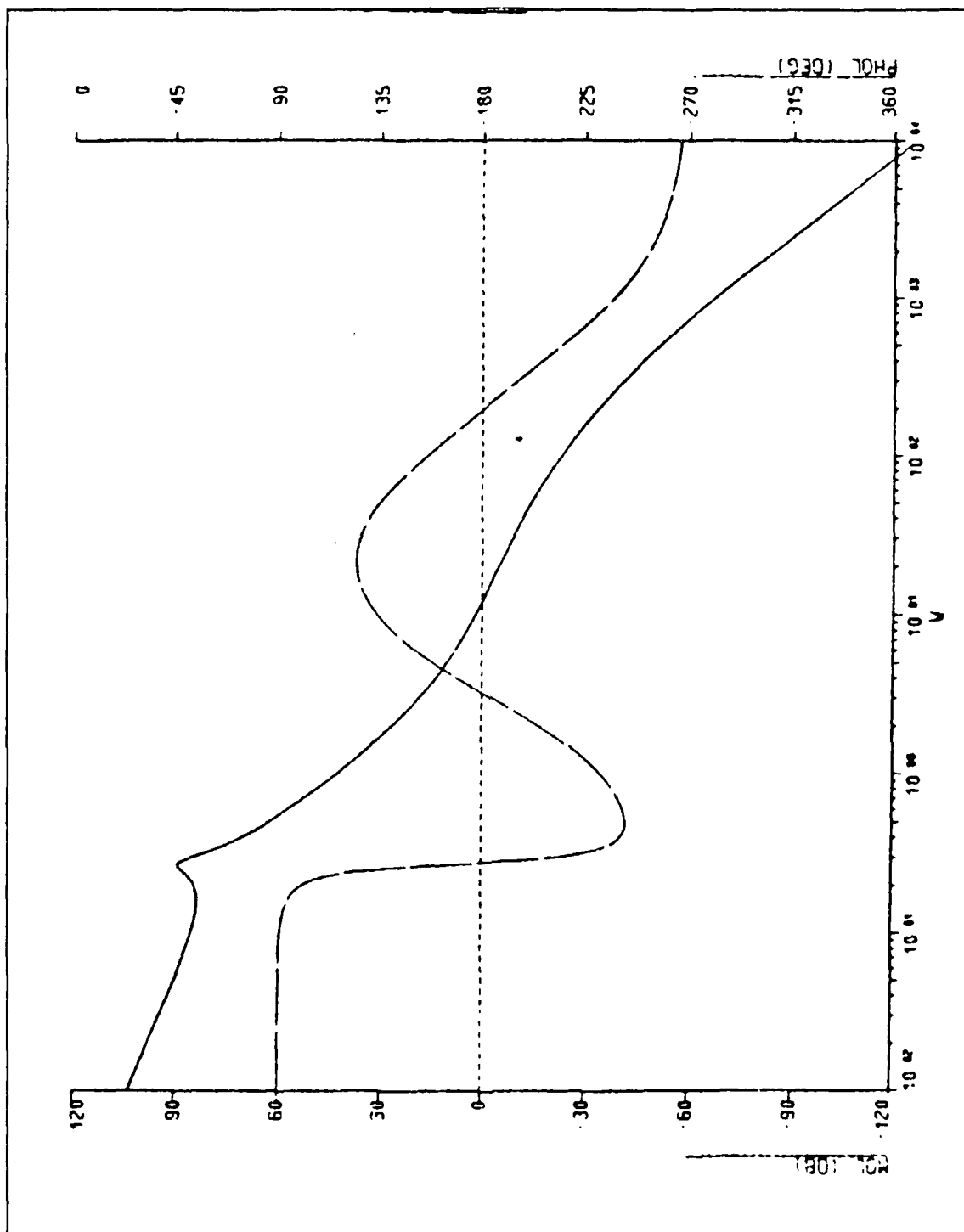


Figure 18. Compensated System Open Loop Bode Plot

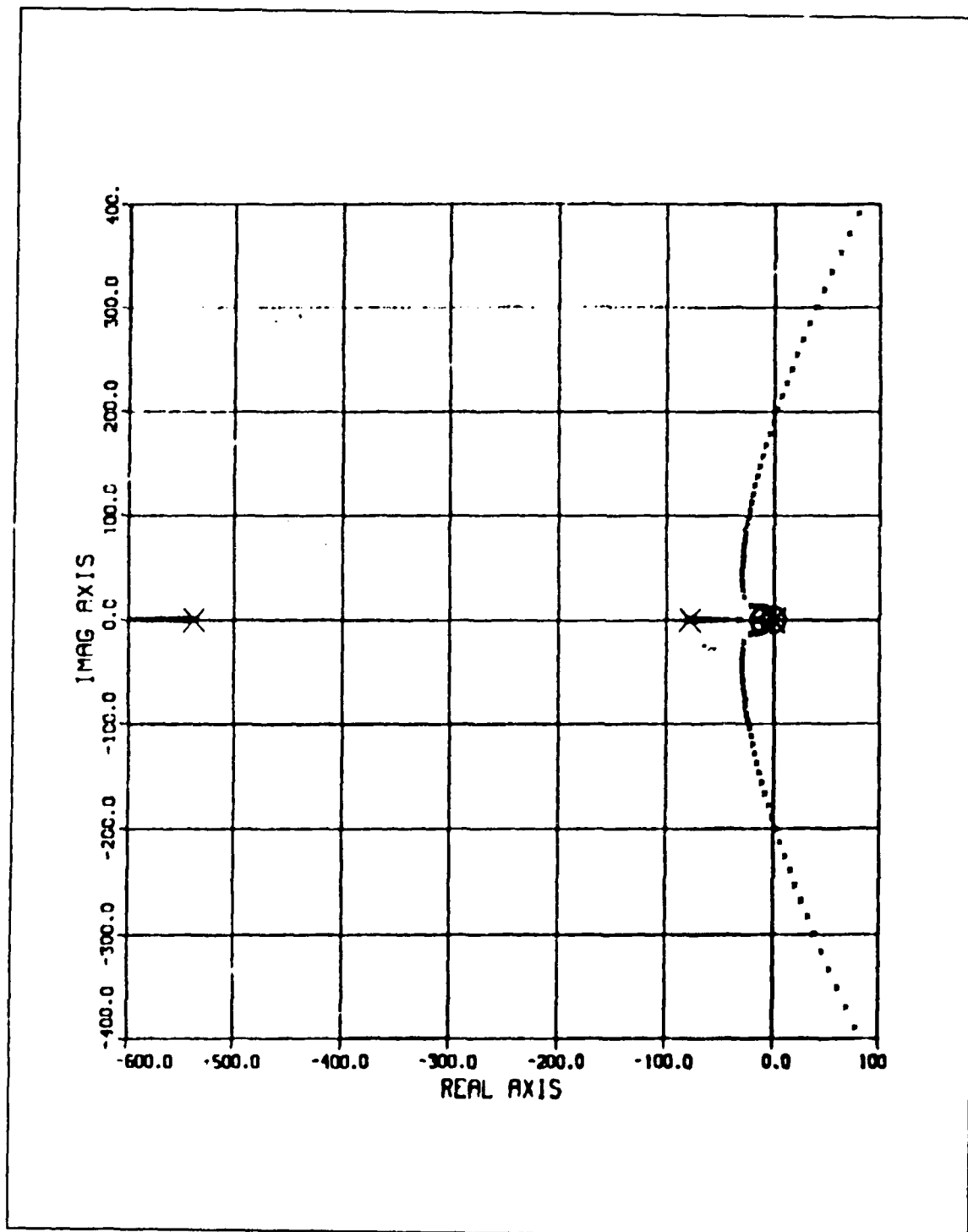


Figure 19. Compensated System Root-Locus

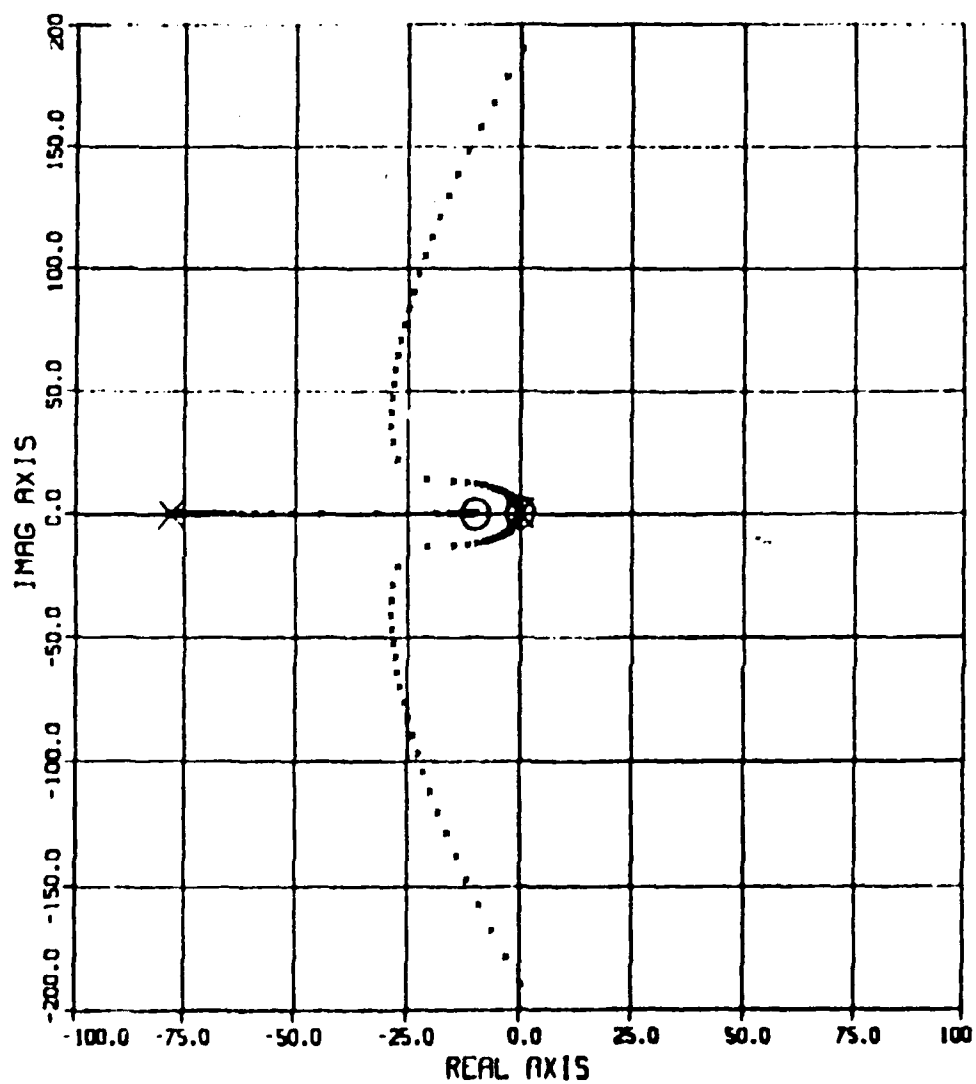


Figure 20. Magnified Root-Locus

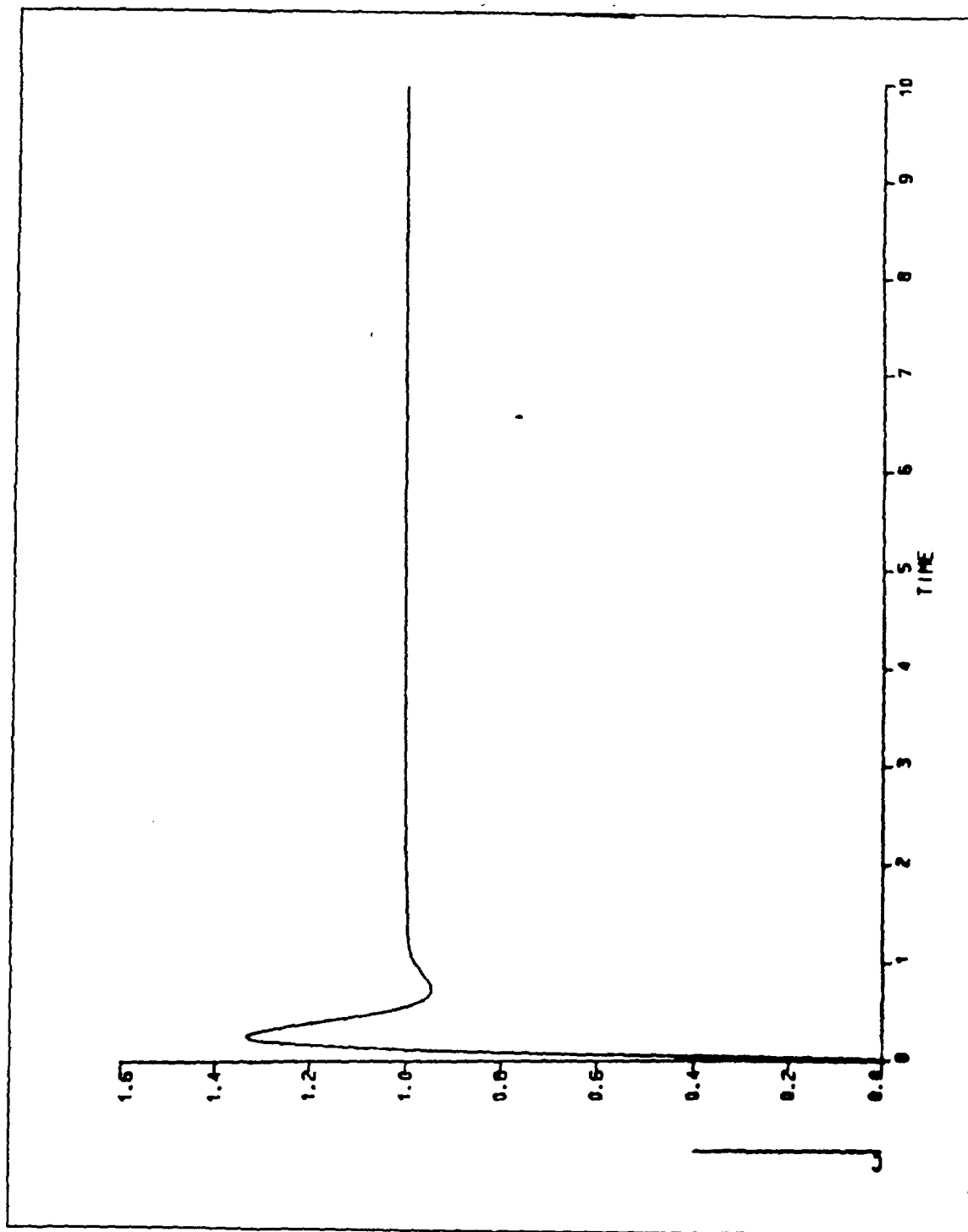


Figure 21. Compensated System Time Response

The second type of manipulation is done to get more information using the Root-Locus method. The first step for this is shown at Figure 22.

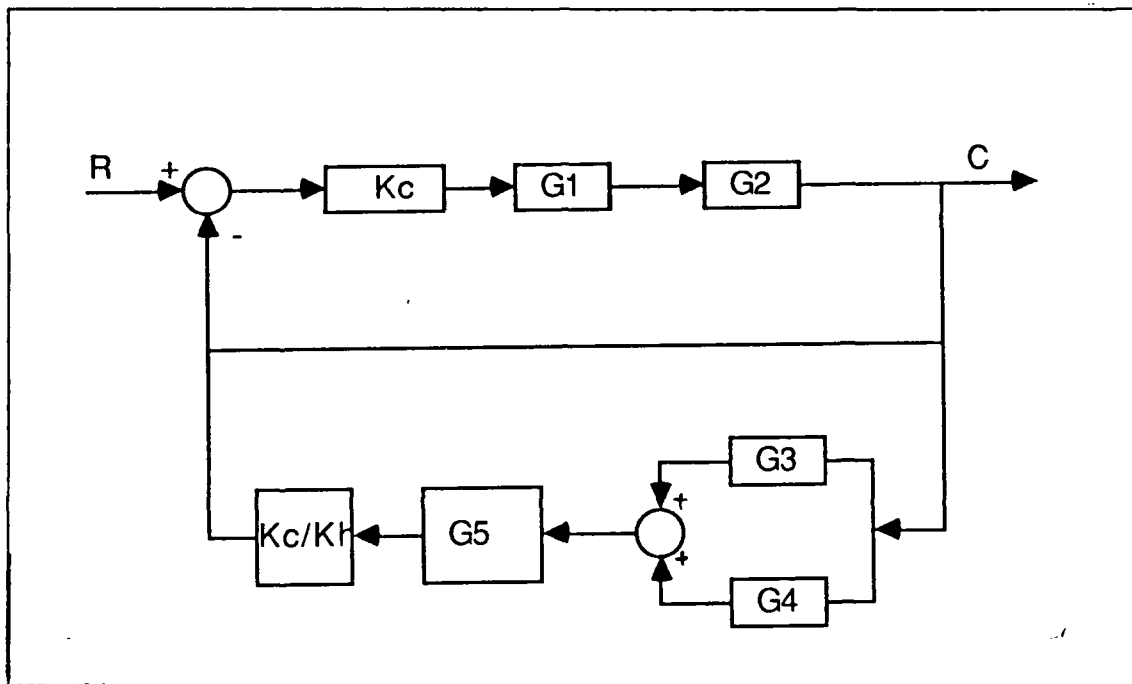


Figure 22. First Step For Manipulation

Then with the appropriate block diagram reduction the system becomes as in Figure 23, where

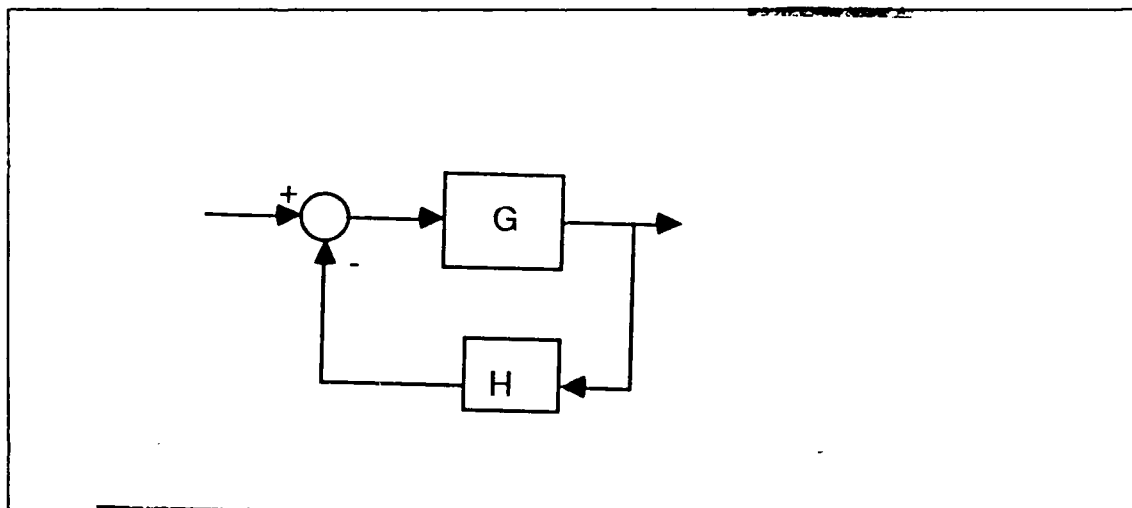


Figure 23. Final System For Root-Locus Examination

$$G = \frac{473898.3}{(s - 26.72 \mp j66.9)(s + 91.74)} \quad (17)$$

$$H = \frac{0.0012s^3(s + 72.39)}{(s + 2.5)(s + 4)} \quad (18)$$

By using the loop transfer function $GH(s)$ to draw the Root-Locus we may examine the system for root movements. Since $G(s)$ has two roots in the right half plane, the Bode plot of this cascade combination cannot be interpreted from the point of view of gain margin or phase margin, but it shows the stability of the system.

The root locus for the uncompensated system is shown in Figure 24. From this point we can proceed to explore the design procedure of the designer.

First of all, since this design was done using Bode design methods, it will be helpful to get the G and $1/H$ magnitude curve which are shown in Figure 25 for the original system in Figure 15.

As we may see from the block diagram there are three zeros at the origin. As in the first example we have to have at least one zero at the origin to keep the system type number unchanged. Also since we want to keep the error coefficient unchanged we have to have one additional zero at the origin. The reason for the third zero is to provide a dipole near the origin to make the system have a very high error coefficient for steady state accuracy and a very narrow bandwidth.

For the $GH(s)$, the Root-Locus and a close look around the origin to the Root-Locus are given in Figures 26 and 27.

Now we may want to see the effect of removing some of the components from the compensator. First, if we remove the third zero the Root-Locus becomes as in Figure 28. The system is unstable. Now we may want to bend the loci by moving one of the poles close to the origin toward left. If we start to move the pole which belongs to the dipole at $s = -2.5$; Figure 29 shows the condition when this pole is at $s = -50$. The system is still unstable. By putting this pole further from the zero at $s = -72.39$, we can bend the loci toward the left half plane. We may interpret this as follows: when we move the pole to the left we increase the sum of the poles and therefore the sum of the roots. This affects the Root-Loci by moving the asymptote centroid to left, so the loci from the right half plane cross into the left half plane. Figure 30 shows the Root-Loci for this condition. Now we may want to examine the specifications of the system

and compare them with our specifications. Since the dominant roots are far away from the origin the accuracy and damping will decrease, and the bandwidth of the system will increase. These do not agree with our specifications.

After all these trials and sample designs we may summarize the results as a set of rules for design of feedback compensation :

1. To keep the system type number unchanged put at least the same number of zeros at the origin as the original system's type number.
2. To keep both the system type number and the error coefficient same unchanged, the number of zeros at the origin should be $N + 1$, where N is the type number of the uncompensated system.
3. To have high error coefficient and narrow bandwidth for steady state accuracy include a dipole close to the origin. Put the zero of the dipole at the origin for ease in physical realization.

In the next chapter we will apply these rules to several different kinds of problems and see whether they work or not.

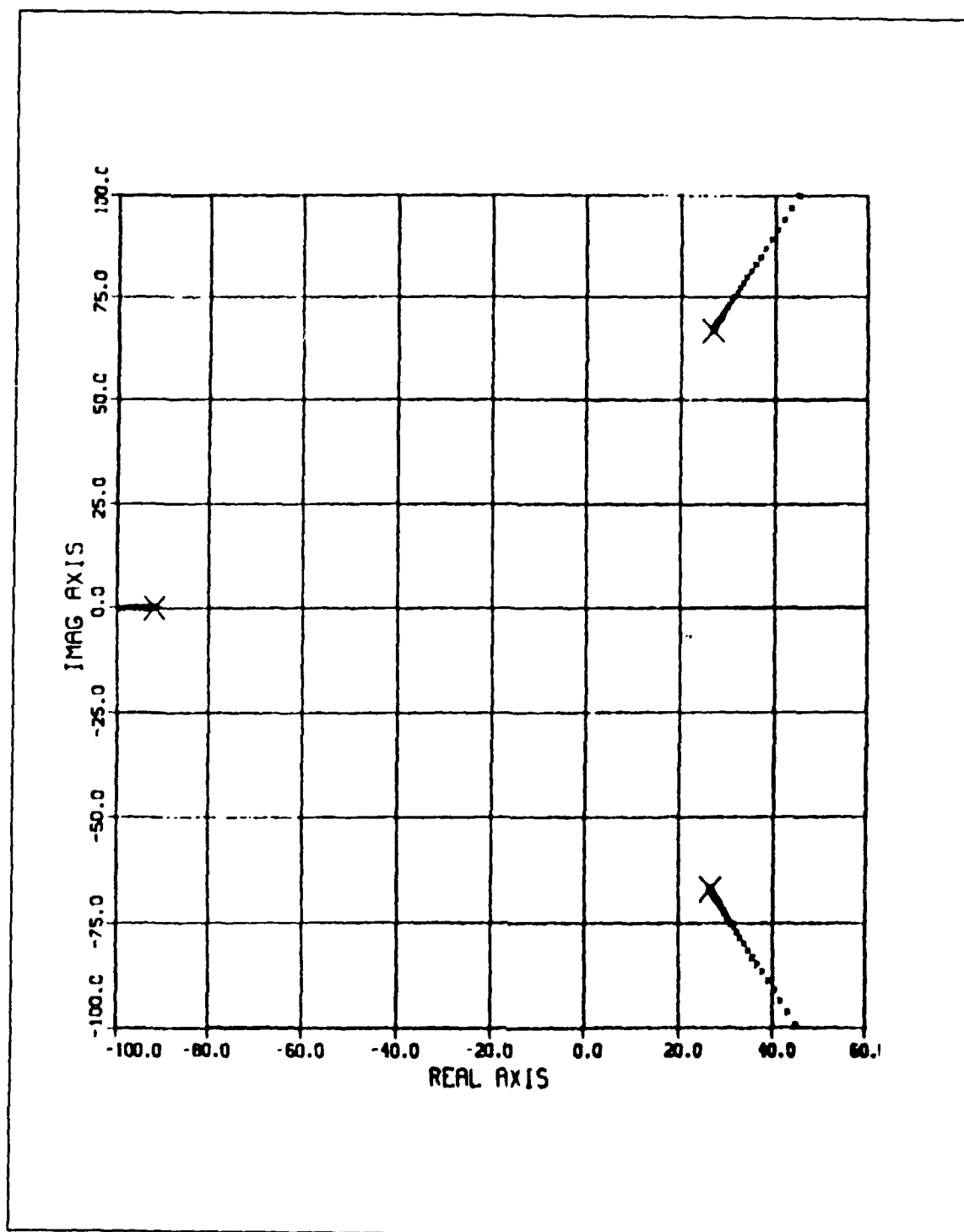


Figure 24. Root-Locus For Uncompensated System

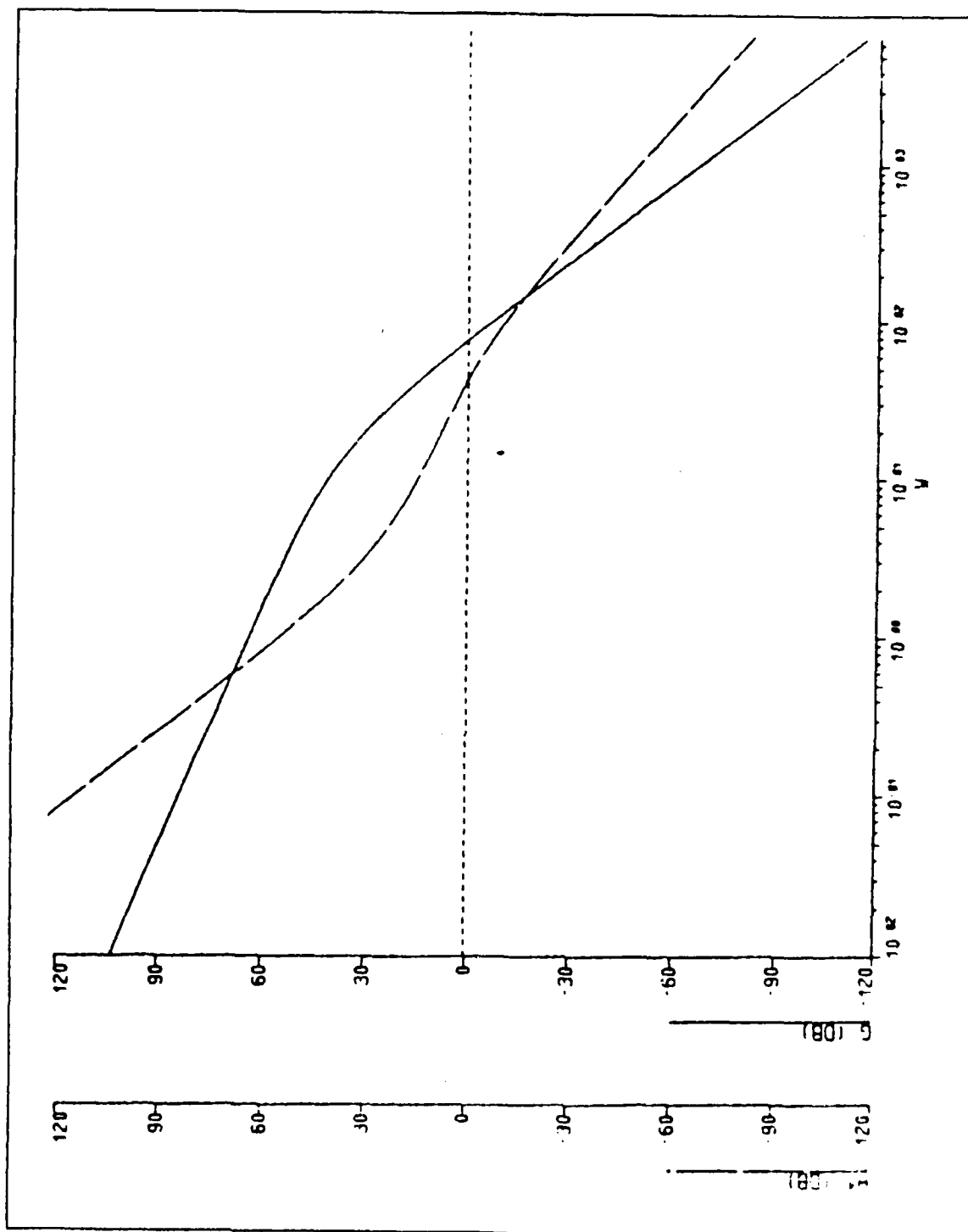


Figure 25. G and $1/H$ Magnitude Curves

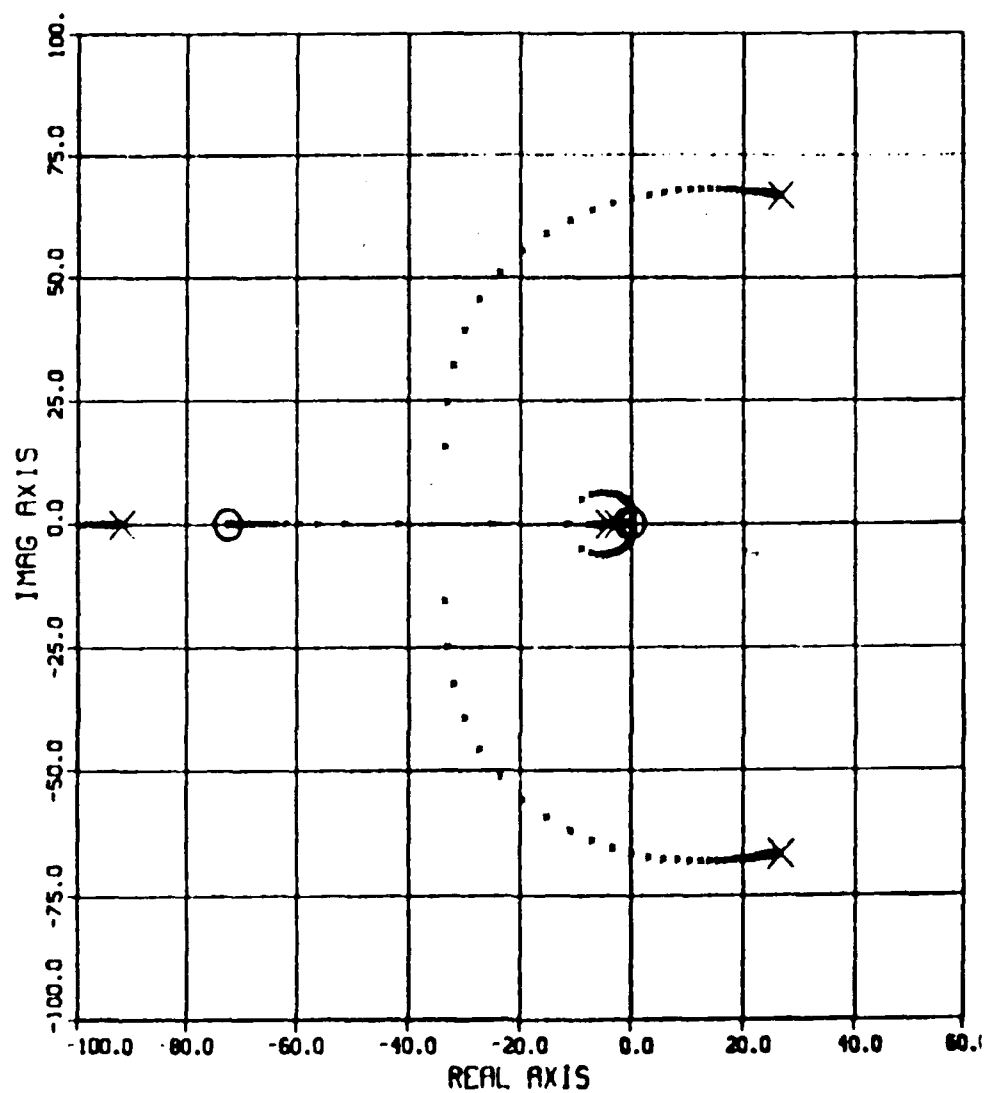


Figure 26. Root-Locus For The Loop Transfer Function $GH(s)$

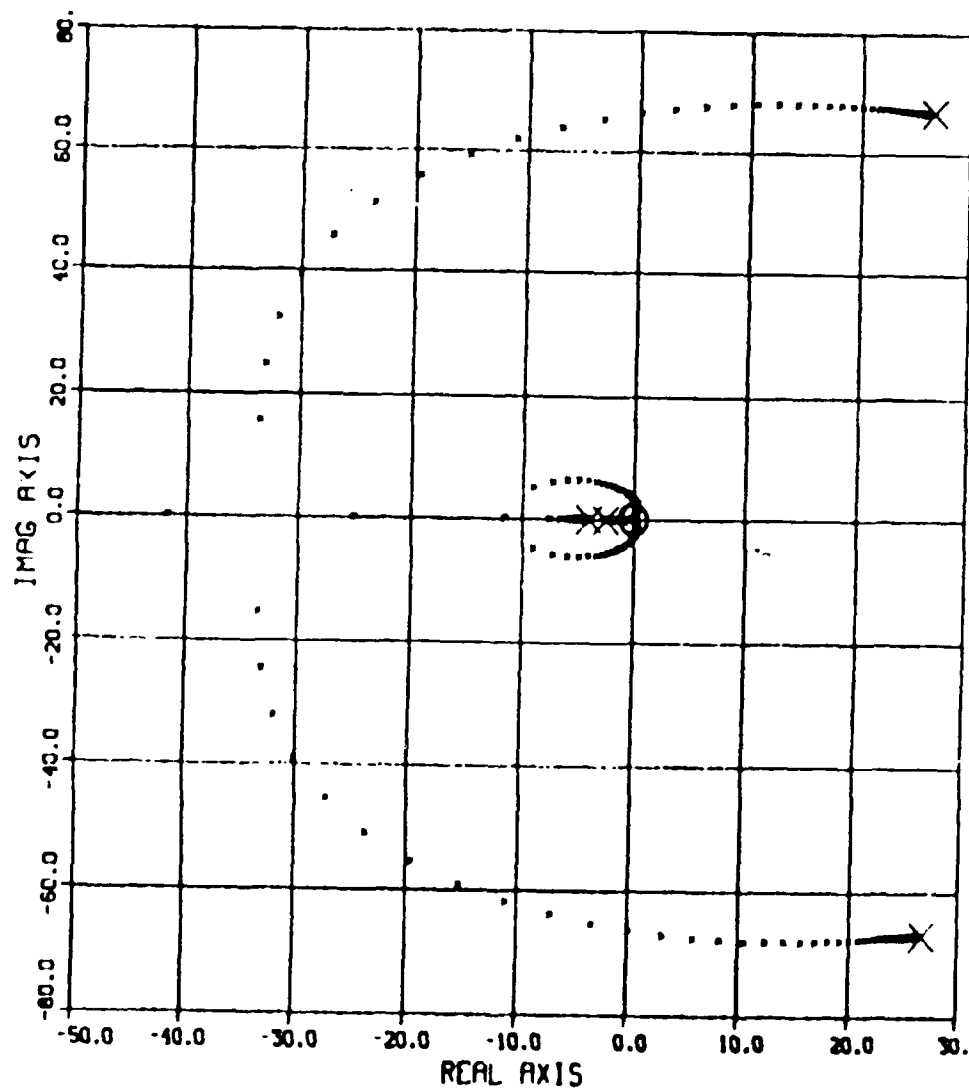


Figure 27. Magnified Root-Locus For The Loop Transfer Function $GH(s)$

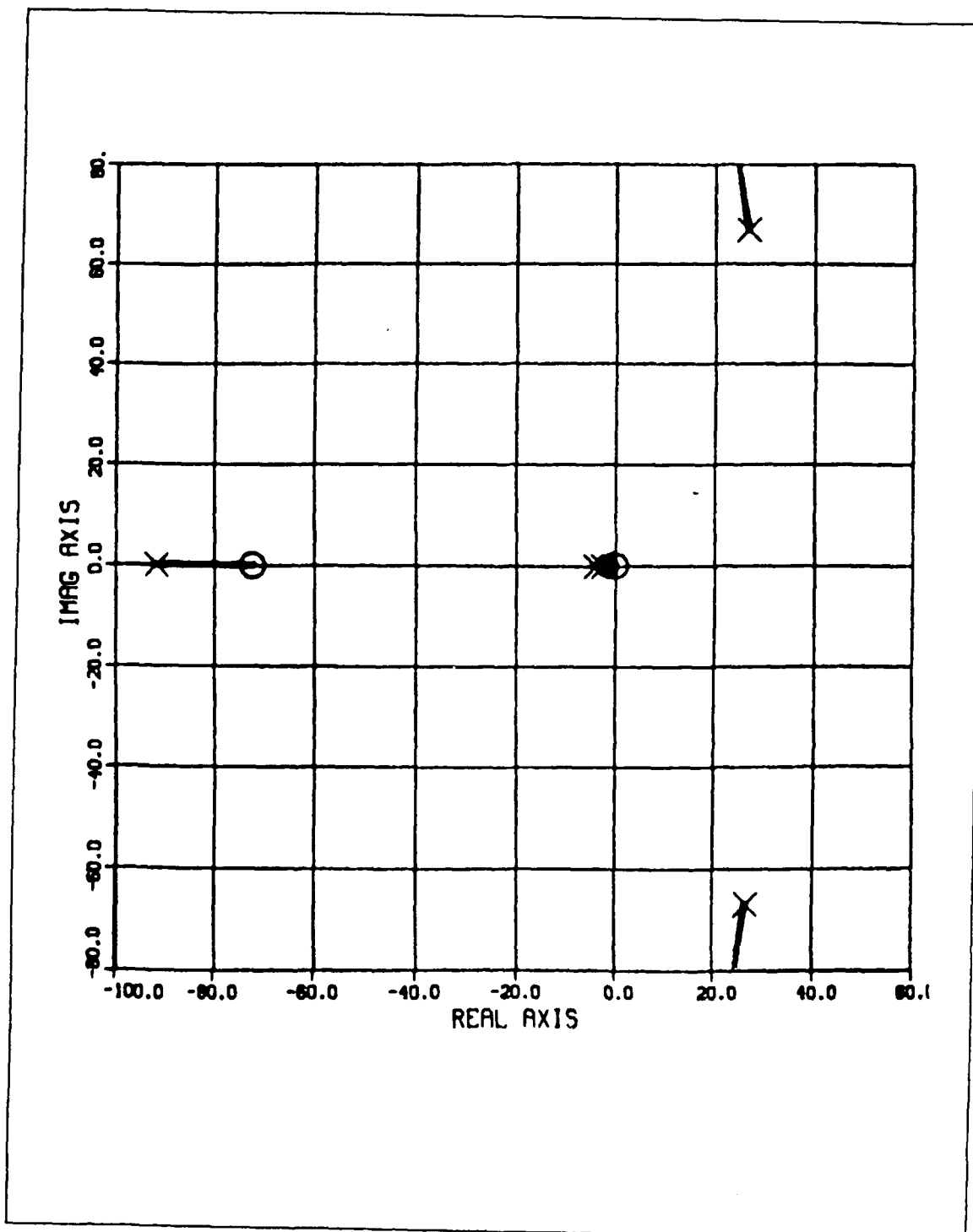


Figure 28. The Same System With Two Zeros at Origin.

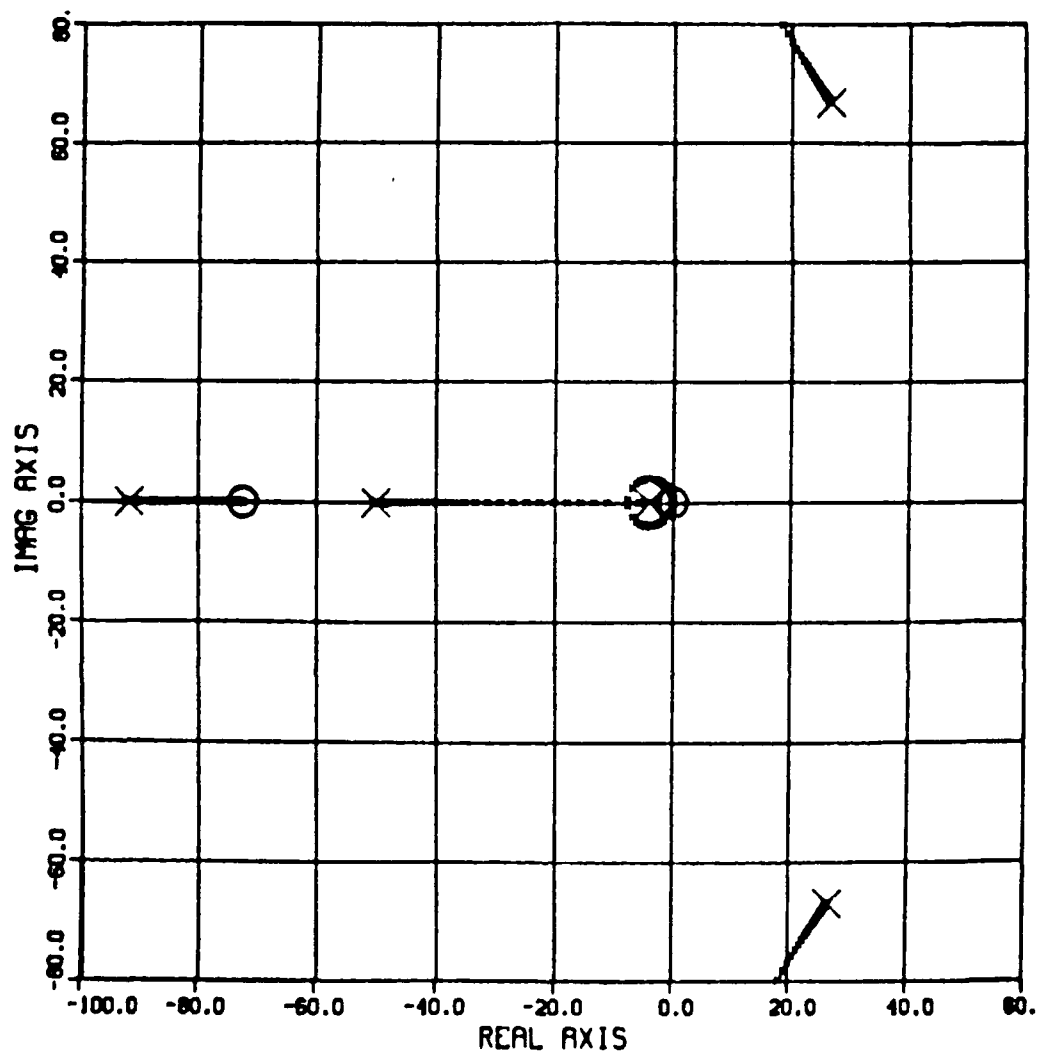


Figure 29. Two Zeros Pole at $s = -50$

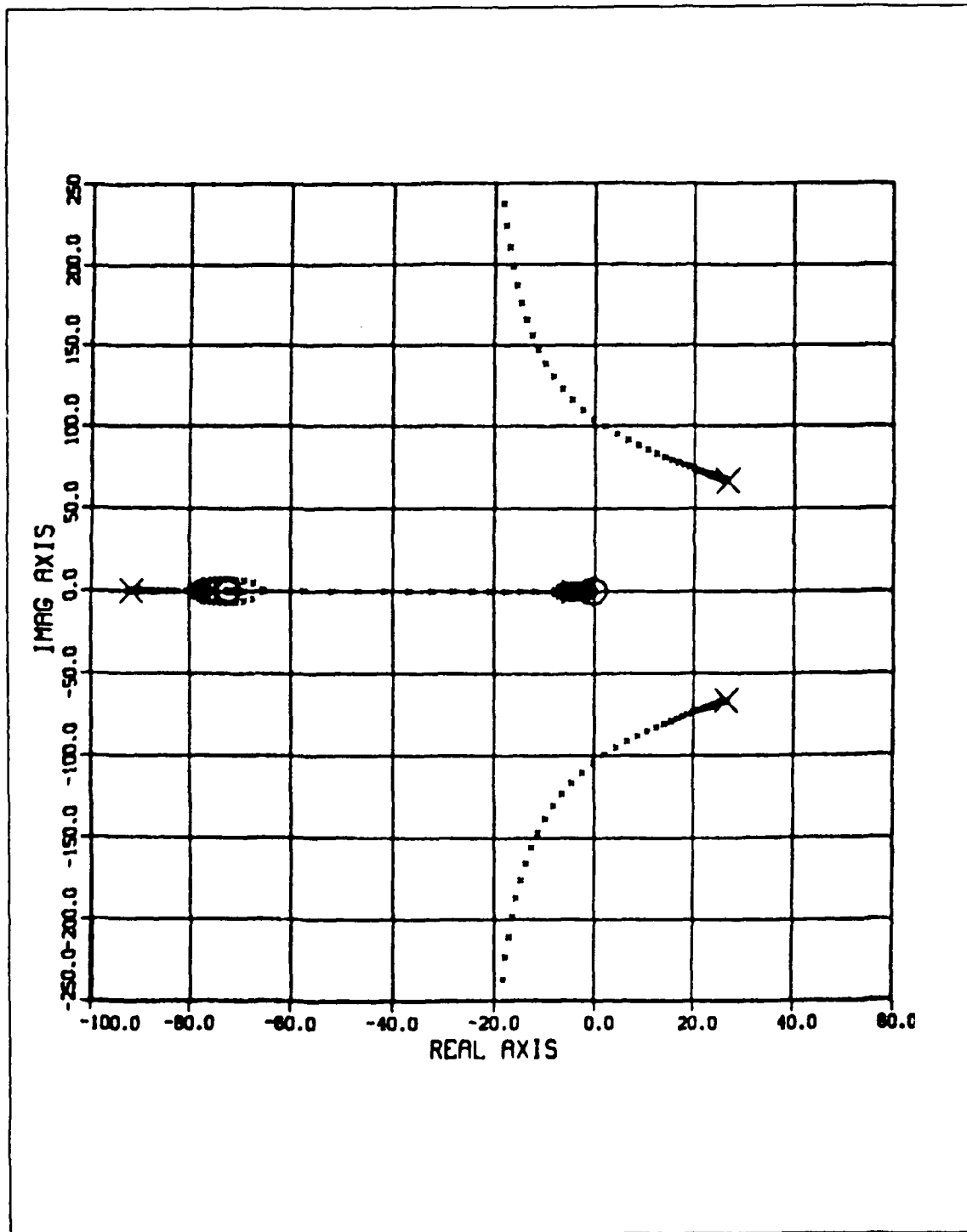


Figure 30. Two Zeros Pole at $s = -75$

III. DESIGNING COMPENSATORS

A. NON-MINIMUM PHASE SYSTEMS

Before studying systems that have all poles and zeros in the left half plane, we may want to examine non-minimum phase systems. If we have a root or roots in the right half plane we are not able to use Bode methods for designing compensators, therefore the Root-Locus method will be the only useful tool.

To illustrate we chose the following transfer function, which is the roll mode of a vertical take-off aircraft. Figure 31 shows the original system, where

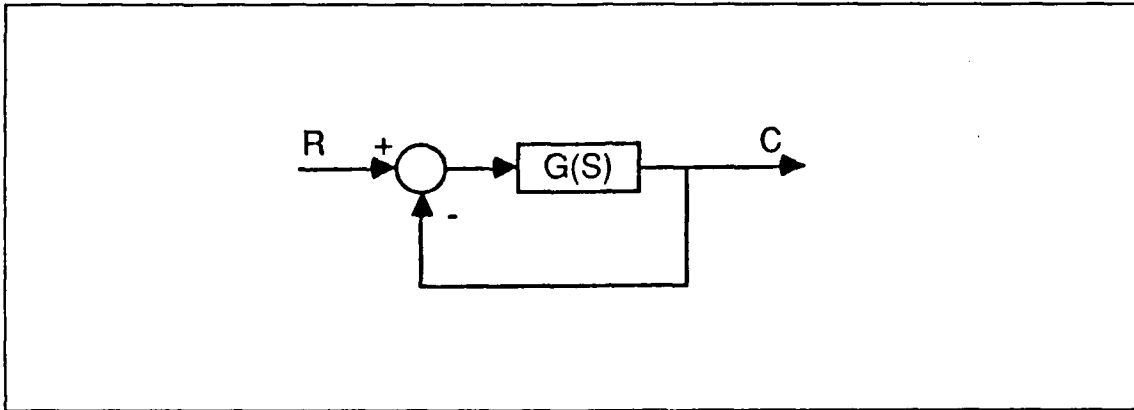


Figure 31. Original System

$$G(s) = \frac{77.187(s + 0.225 \pm j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \pm j0.844)(s + 48.35 \pm j49.34)} \quad (19)$$

As we may see from the transfer function, there is a real pole in the right half plane. For this reason we can not use Bode methods to compensate the system. The open loop Bode plot, Root-Locus and magnified Root-Locus for this system are given by Figures 32 - 34.

To find the closed loop roots we make the block diagram manipulations. Then the closed loop transfer function becomes :

$$G_1(s) = \frac{77.187(s + 0.225 \pm j0.6607)}{(s - 0.036)(s + 0.21 \pm j0.84)(s + 1.474)(s + 48.34 \pm j48.72)} \quad (20)$$

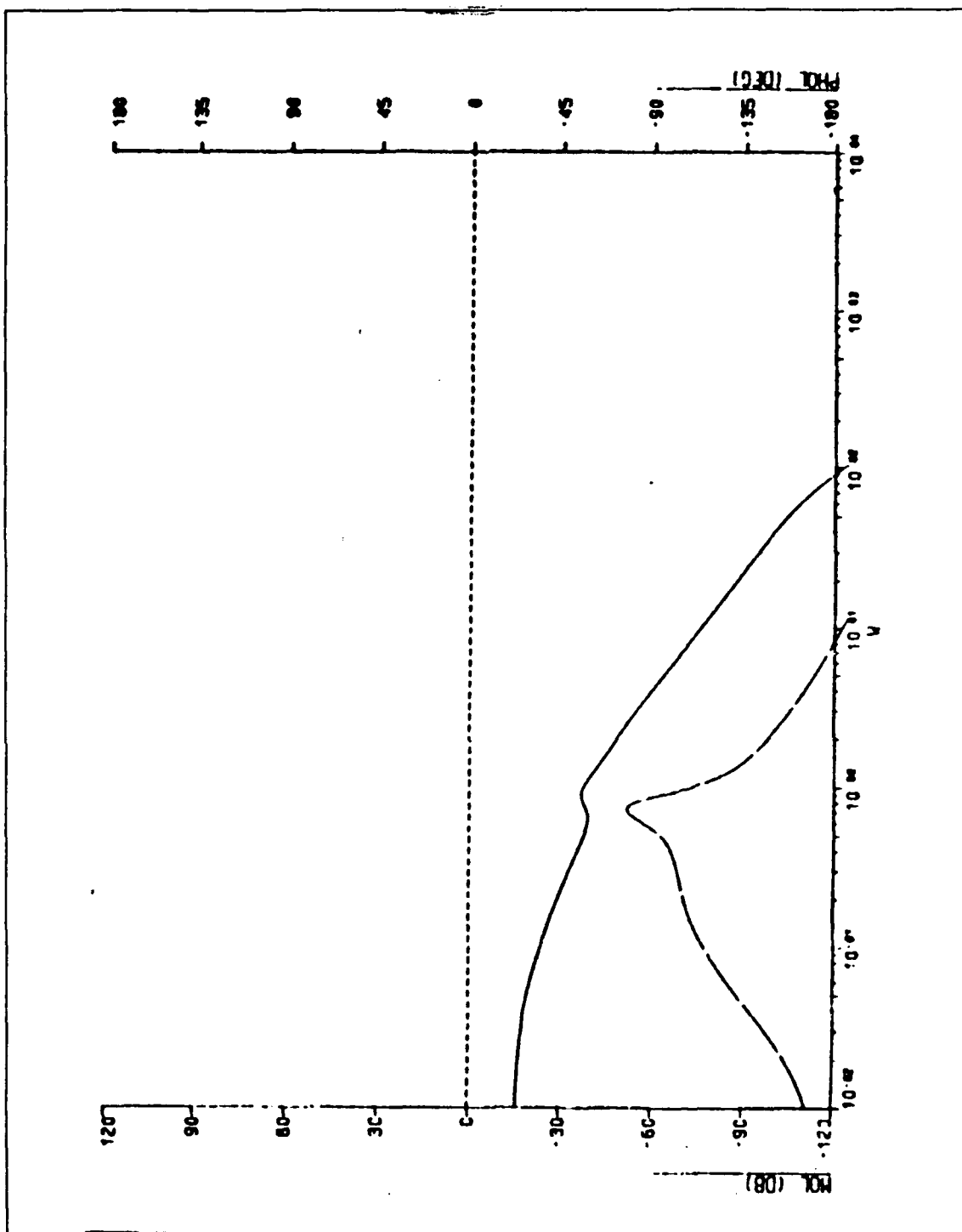


Figure 32. Uncompensated System Open Loop Bode

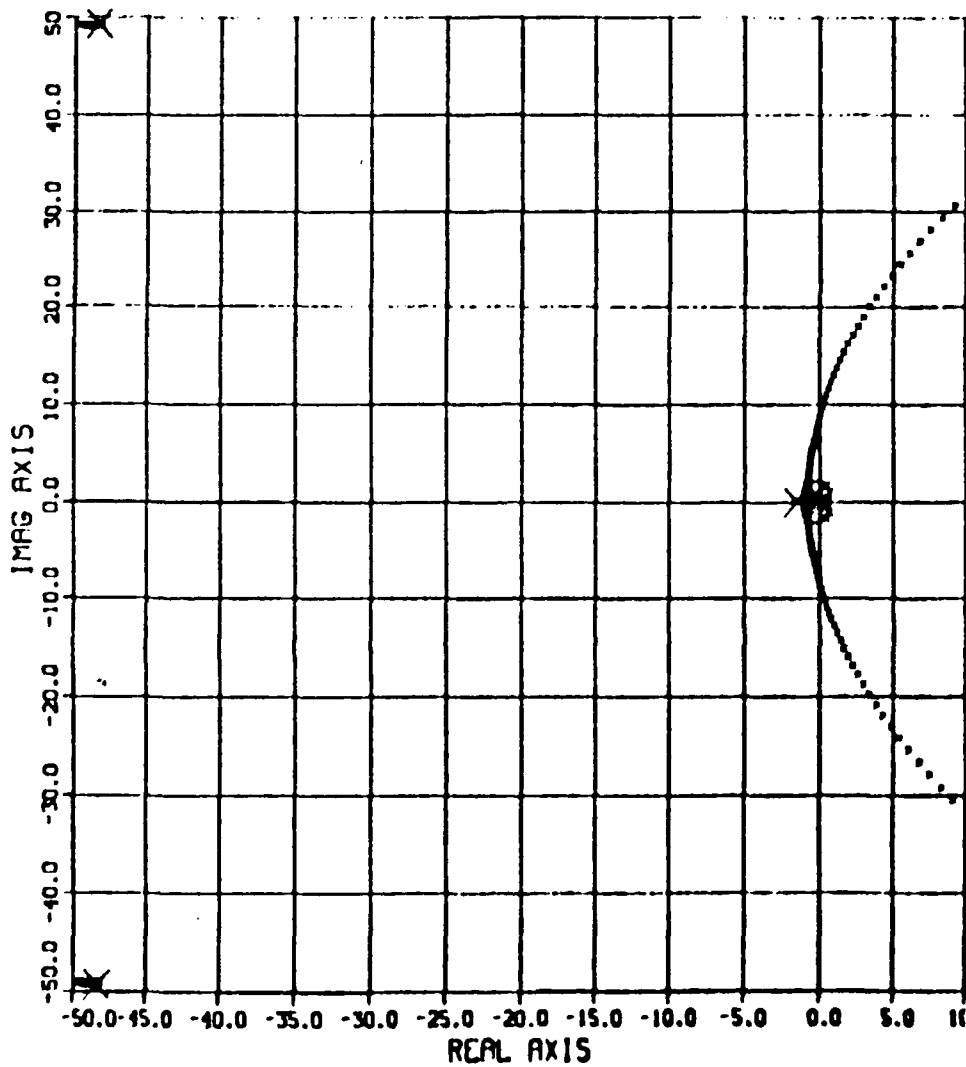


Figure 33. Root-Locus For Uncompensated System

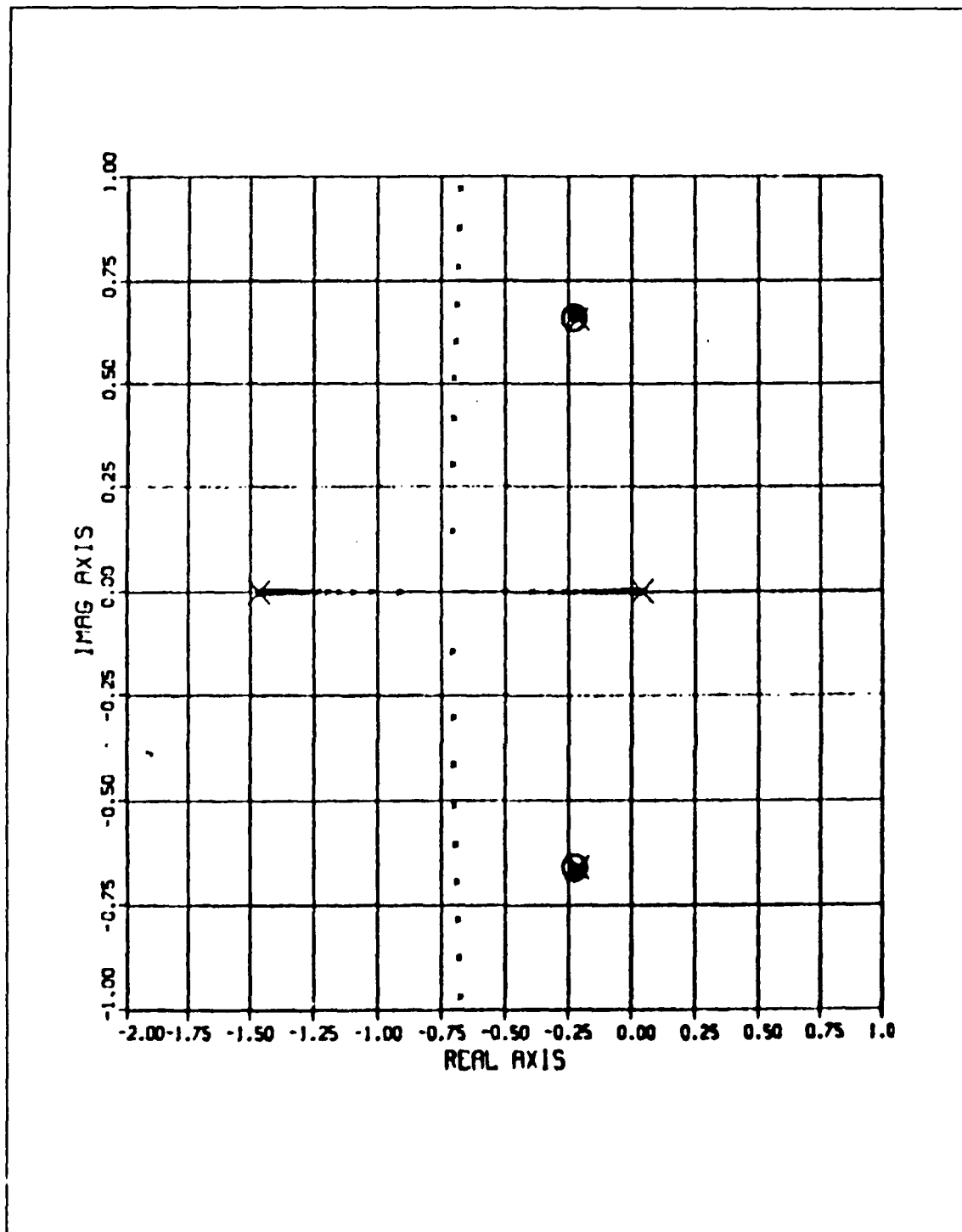


Figure 34. Magnified Root-Locus For Uncompensated System

From this point on we may start to think in terms of Root-Locus and loop transfer functions.

Since we will want to have loci from the pole in the right half plane toward the left, we have to put either a zero or a pole to the left of this pole. Putting a zero at the origin will not help us, because in that case we always will have a root in the right half plane. So we should put the compensator poles and zeros somewhere in the left half plane. Choosing the specific zero and pole locations is an engineering decision and mostly depends on the specifications. With this problem we will try to illustrate three different combinations.

First put a zero close to the origin and a pole far away, both in the left half plane. In this case the loop transfer function becomes as in Figure 35, where

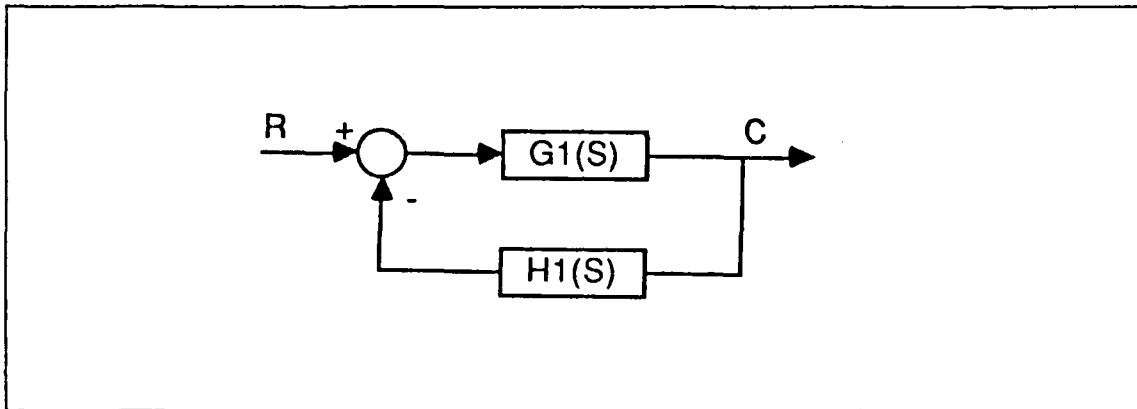


Figure 35. Loop Transfer Function Form

$$G_1(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.036)(s + 0.21 \mp j0.84)(s + 1.474)(s + 48.34 \mp j48.72)} \quad (21)$$

$$H_1(s) = \frac{K_1(s + 1)}{(s + 50)} \quad (22)$$

The Root-Locus and magnified Root-Locus for the loop transfer function are given in Figures 36 and 37. As we may see from the plots there is a root-locus segment between the pole in the right half plane and the zero at -1.

To find the value for K_1 , we should get the tabulated points for the Root-Locus and see for which gain all of the roots are in the left half plane. For this particular filter the minimum value for K_1 was 203.7. We selected $K_1 = 500$ arbitrarily. For this value, our system becomes as in Figure 38, where

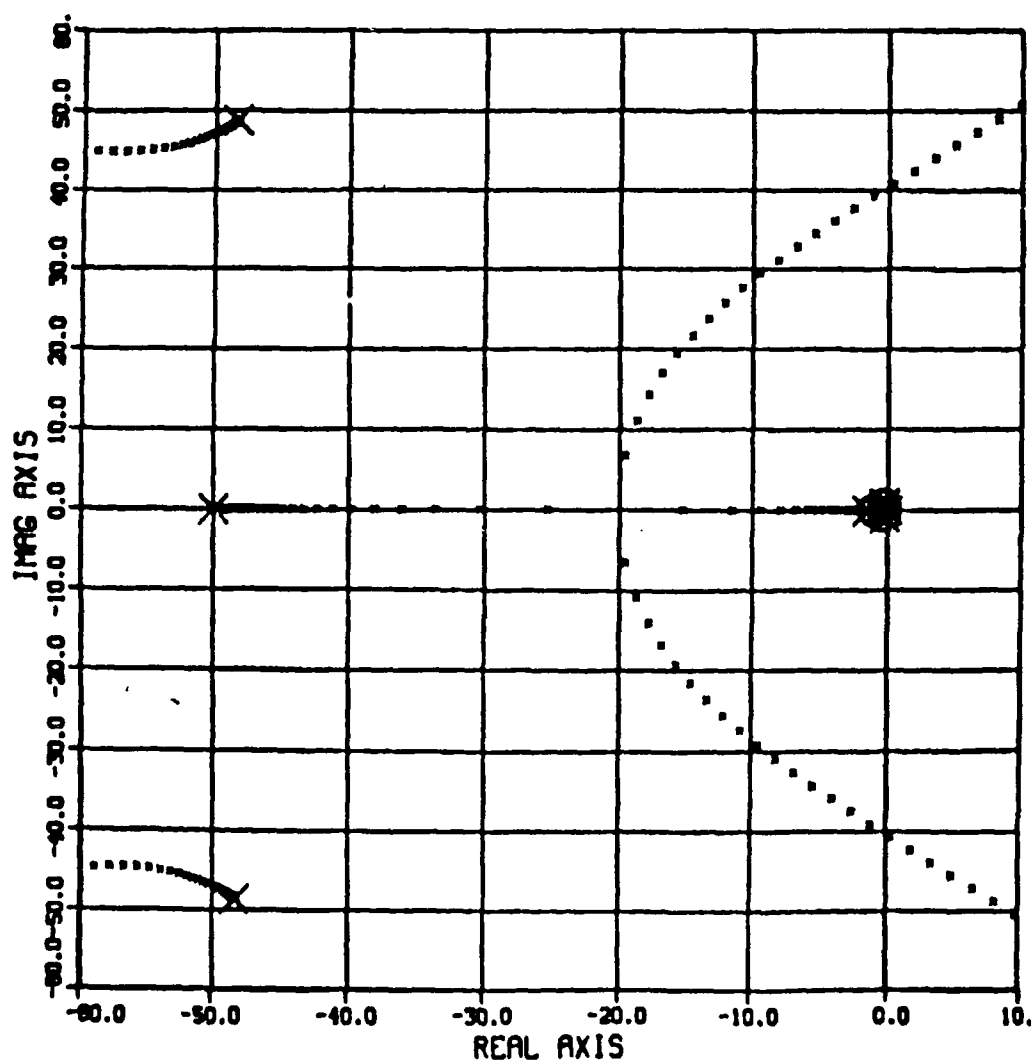


Figure 36. Root-Locus For Loop Transfer Function

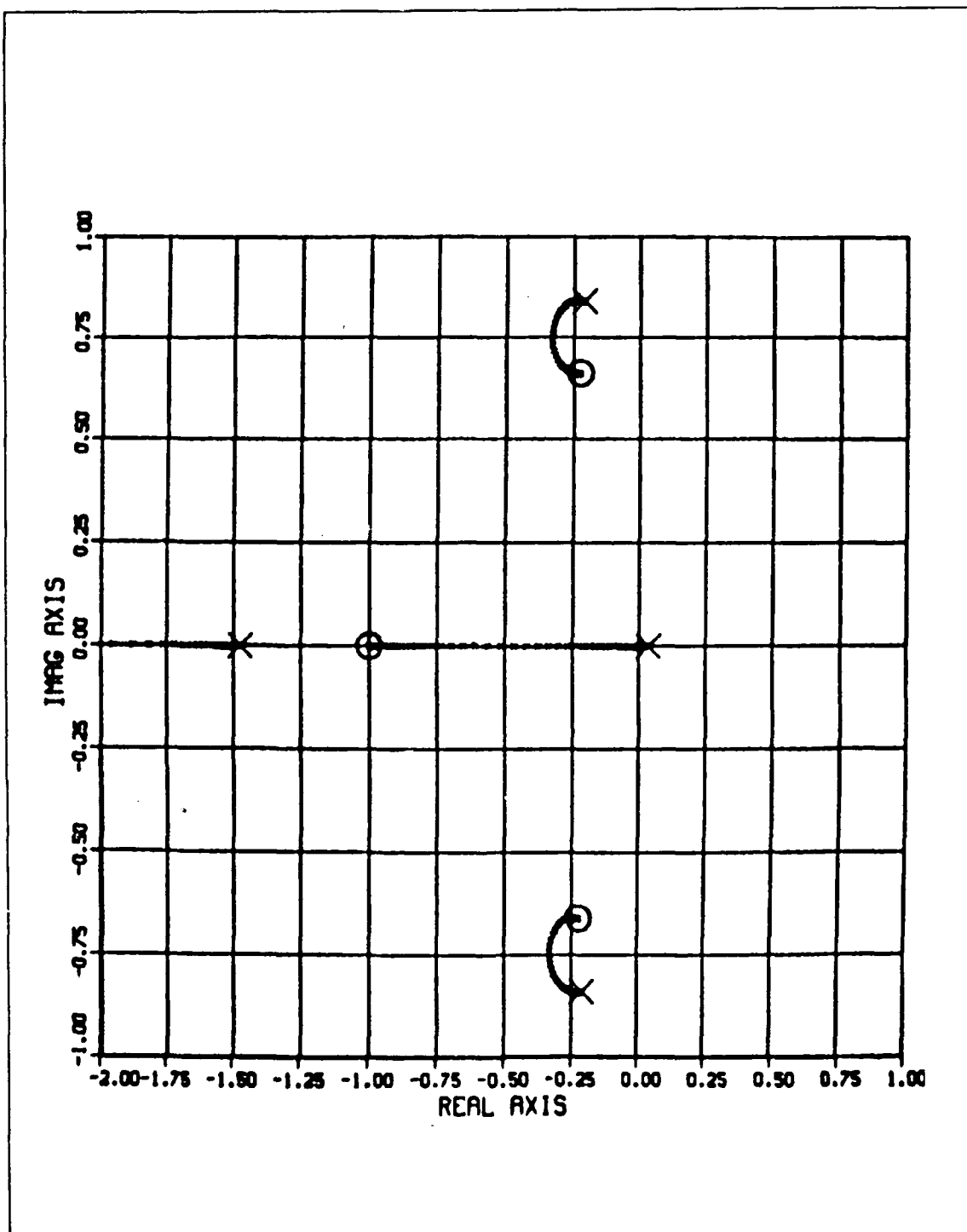


Figure 37. Magnified Root-Locus For Loop Transfer Function

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (23)$$

$$H_1(s) = \frac{500(s + 1)}{(s + 50)} \quad (24)$$

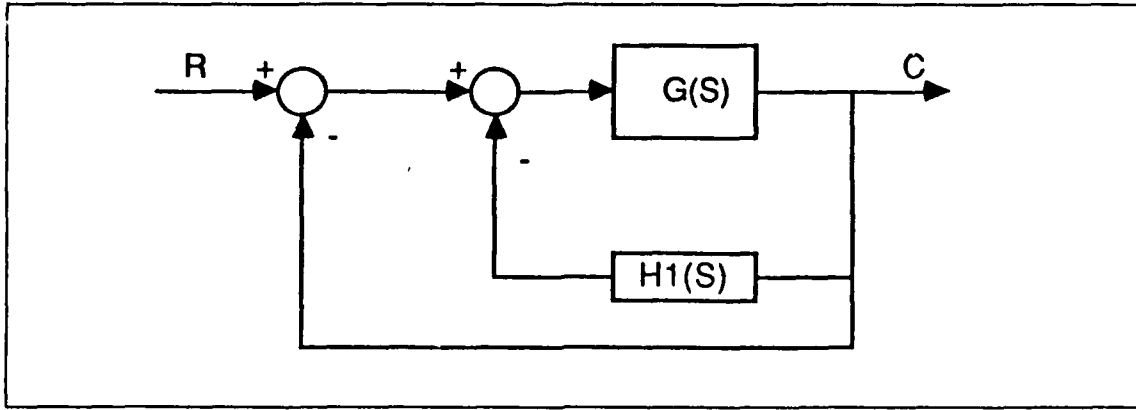


Figure 38. Compensated System

If we make the block diagram manipulation, we get the system shown in Figure 39, where

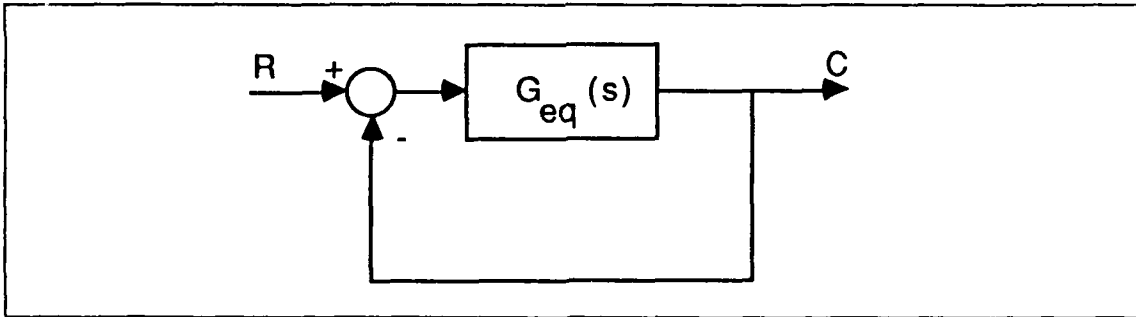


Figure 39. Resulting Equivalent G

$$G_{eq}(s) = \frac{77.187(s + 0.225 \mp j0.6617)(s + 50)}{(s + 0.0269)(s + 0.235 \mp j0.842)(s + 1.535)(s + 49.67)(s + 48.41 \mp j48.63)} \quad (25)$$

The Root-Locus, magnified Root-Locus, open loop Bode plot and time response for the compensated system are given in Figures 40 - 43. As we may see from the Bode plot the system is stable but we do not have enough gain.

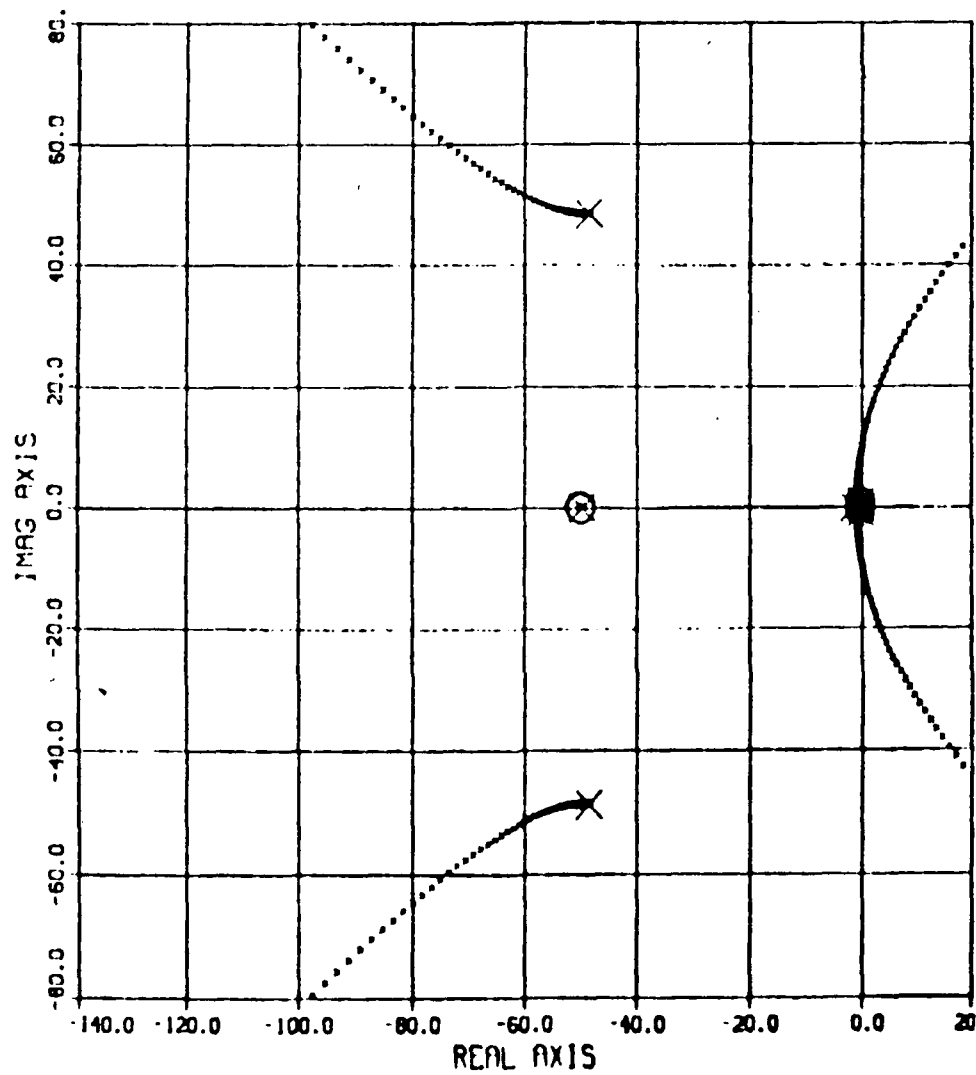


Figure 40. Root-Locus For Final System

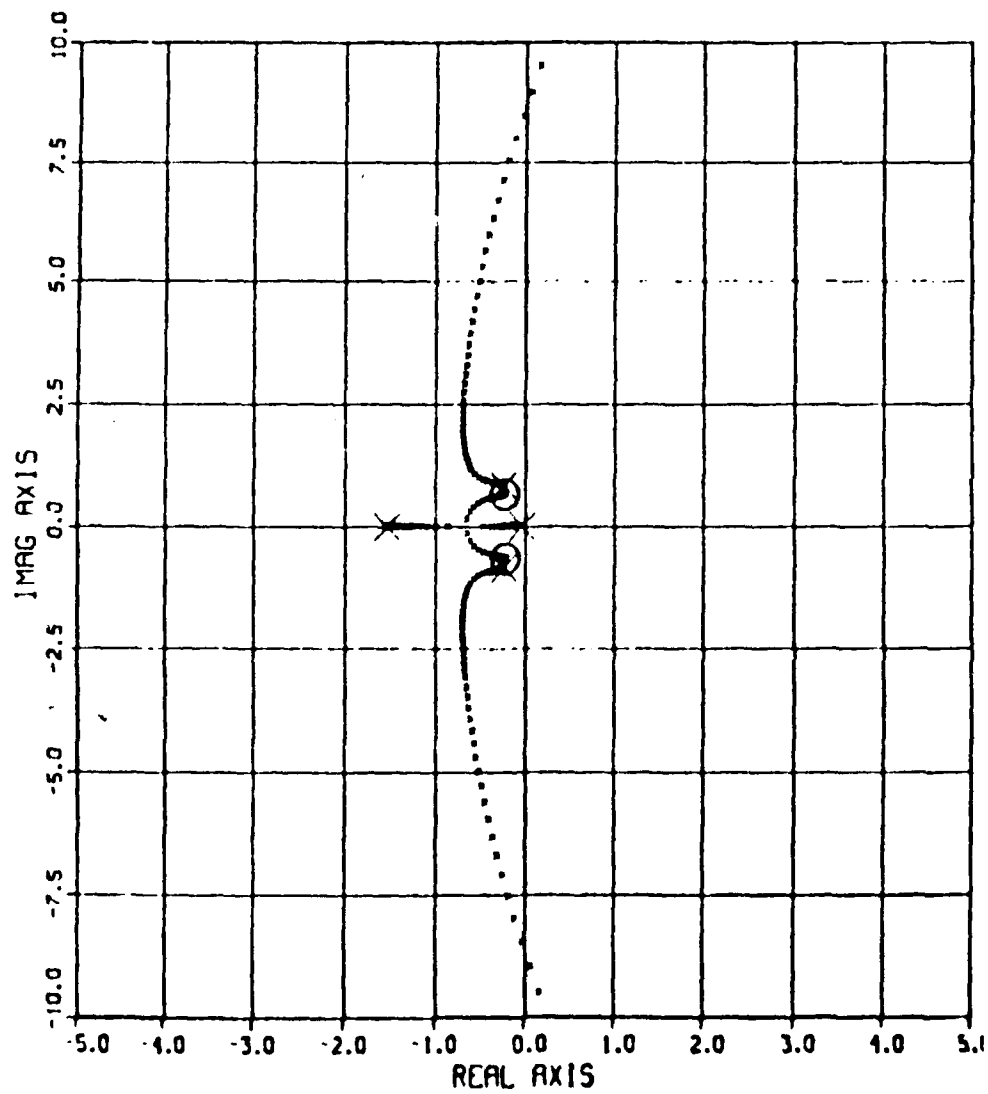


Figure 41. Magnified Root-Locus For Final System

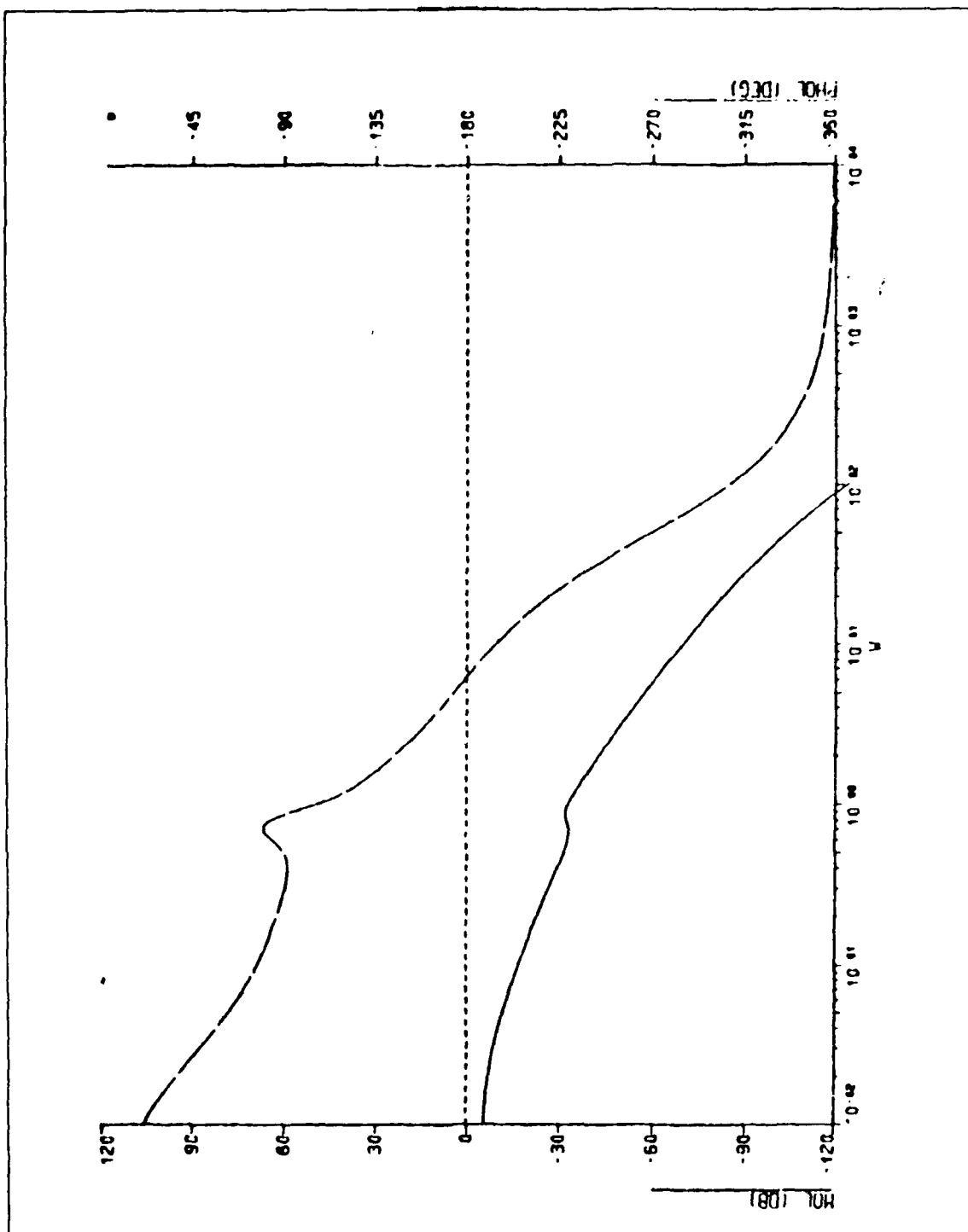


Figure 42. Bode Plot Without Outside Gain

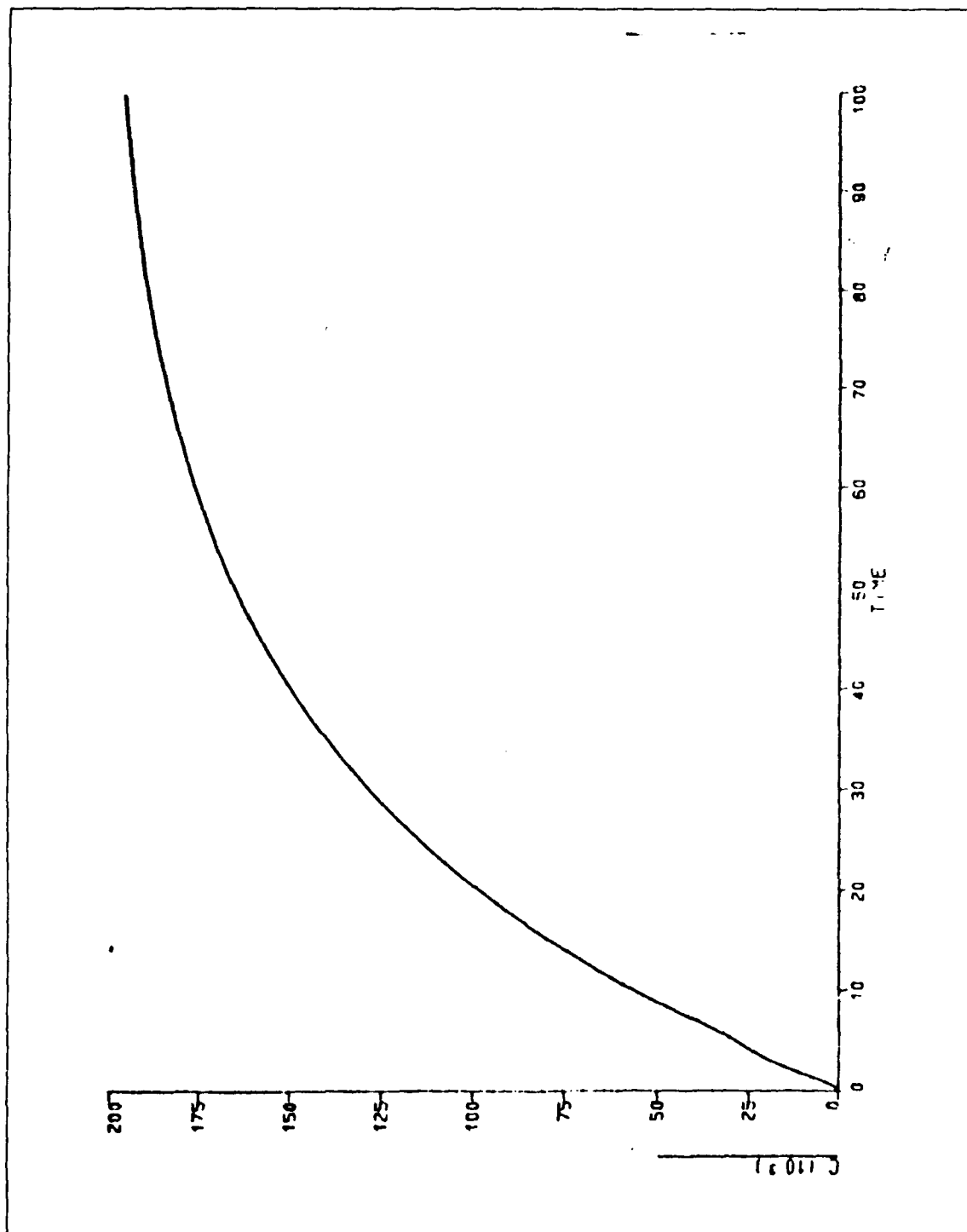


Figure 43. Time Response Without Outside Gain

We may handle this by putting another gain block outside the minor loop. The final system then becomes as in Figure 44.

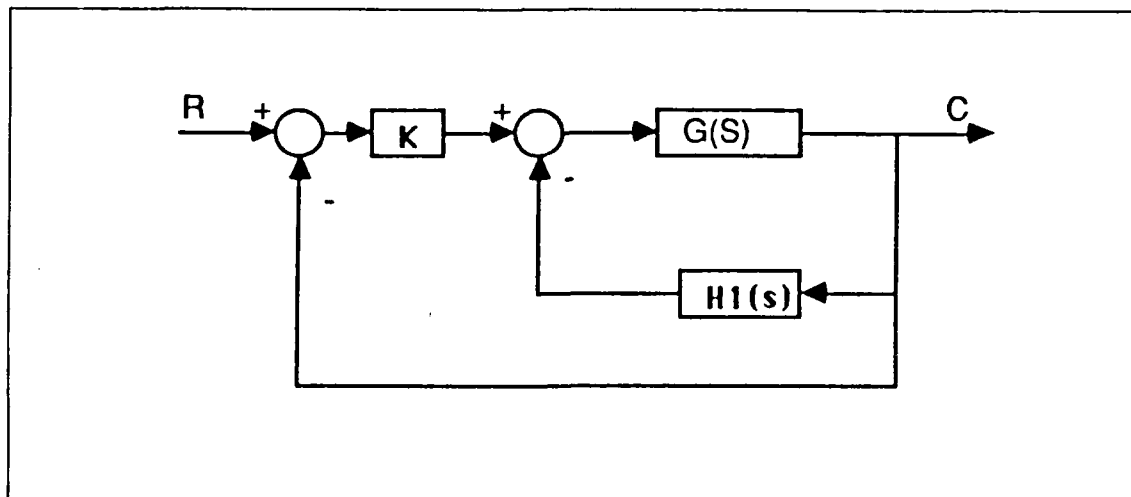


Figure 44. Final Compensated System

K may be selected according to the specifications. We chose $K = 100$ for illustration.

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (26)$$

$$H_1(s) = \frac{500(s + 1)}{(s + 50)} \quad (27)$$

The open loop Bode plot and time response for this particular values are given in Figures 45 and 46. We may change the loop gain according to the specifications. Effects of changing the loop gain will be shown for the third combination.

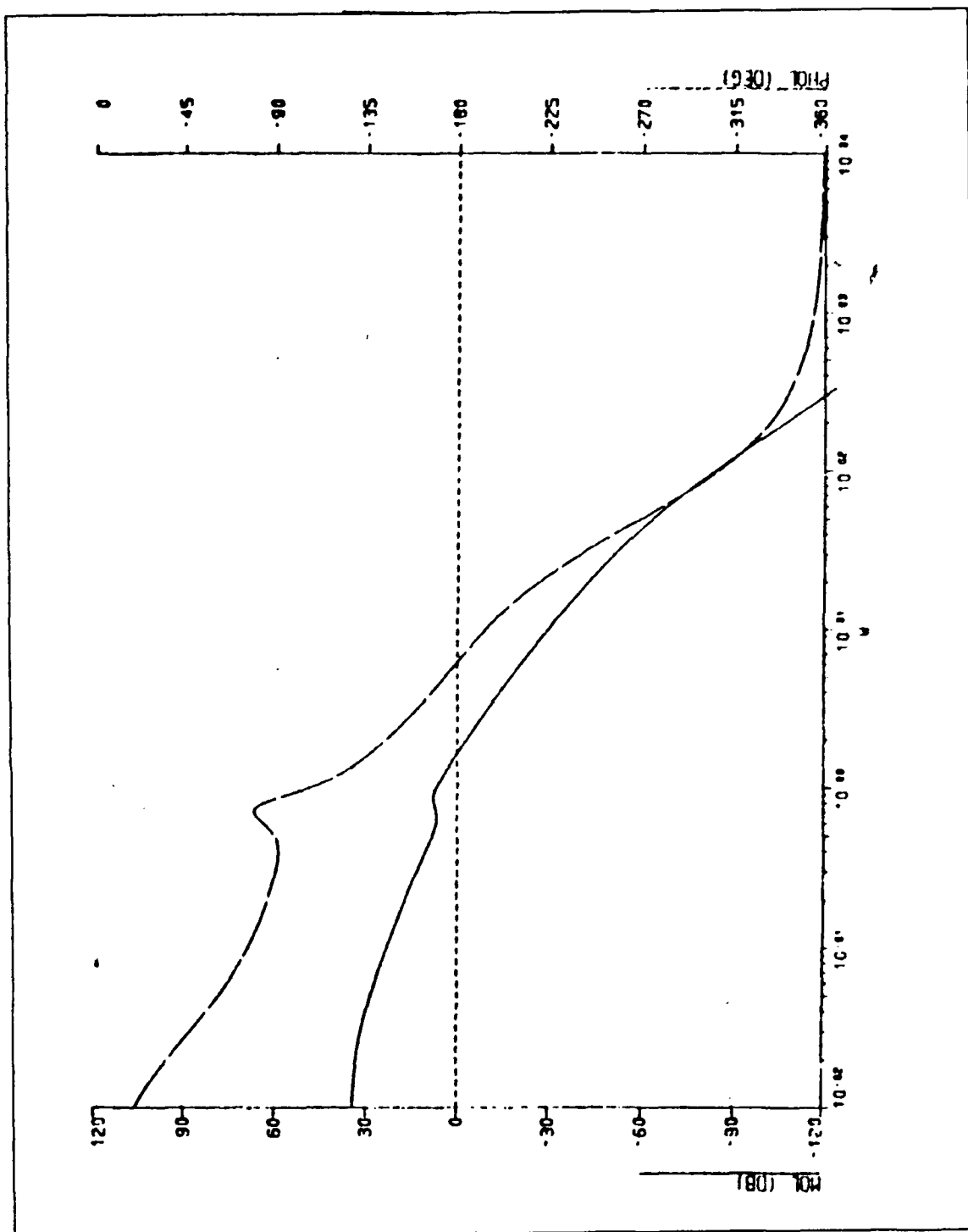


Figure 45. Bode Plot For Final System Loop Gain = 100

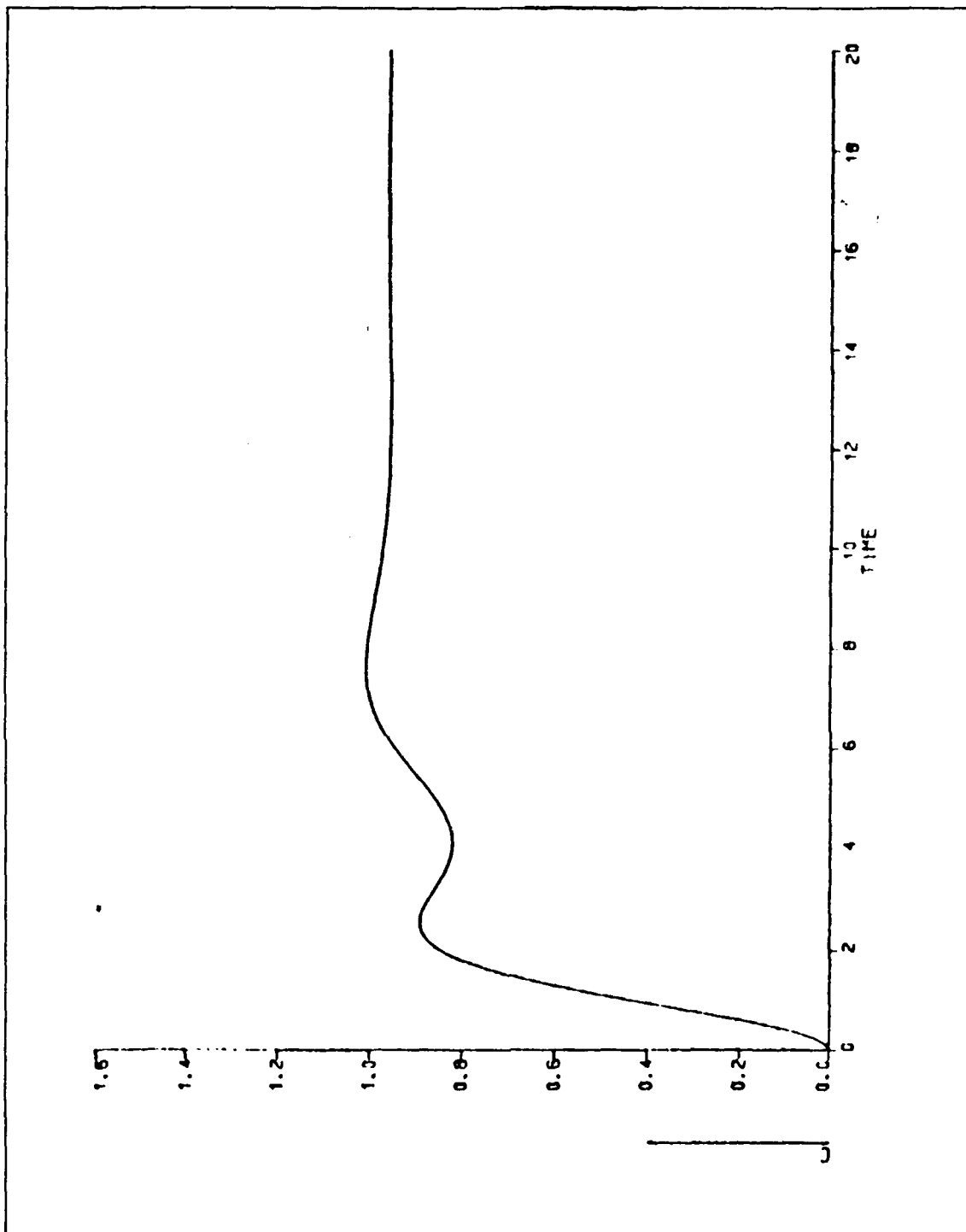


Figure 46. Final System Loop Gain = 100

The second type of configuration puts two complex zeros close to the origin and two poles far away, all in the left half plane. In this case the loop transfer function becomes as in Figure 47, where

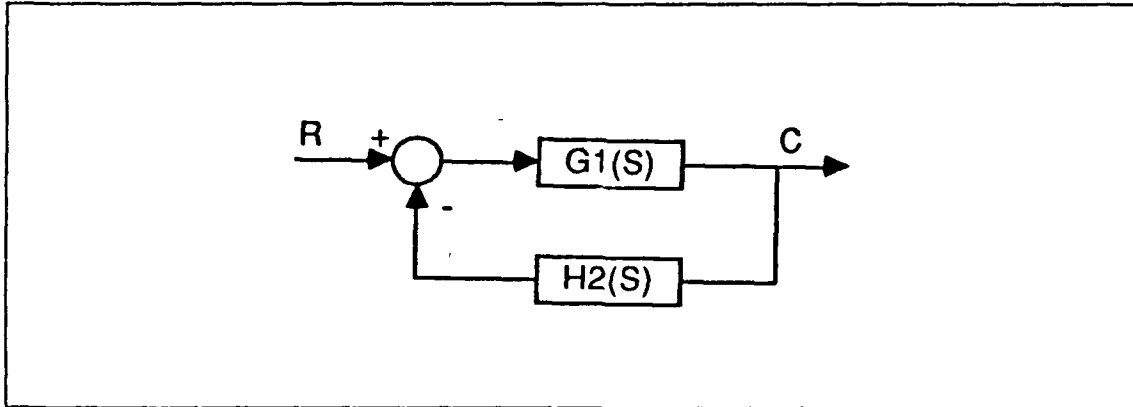


Figure 47. Loop Transfer Function Form

$$G_1(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.036)(s + 0.21 \mp j0.84)(s + 1.474)(s + 48.34 \mp j48.72)} \quad (28)$$

$$H_2(s) = \frac{K_2(s + 1 \mp j)}{(s + 20)(s + 30)} \quad (29)$$

The Root-Locus and magnified Root-Locus for the loop transfer function are given in Figures 48 and 49. As we can see from the plots we moved the loci toward the left half plane.

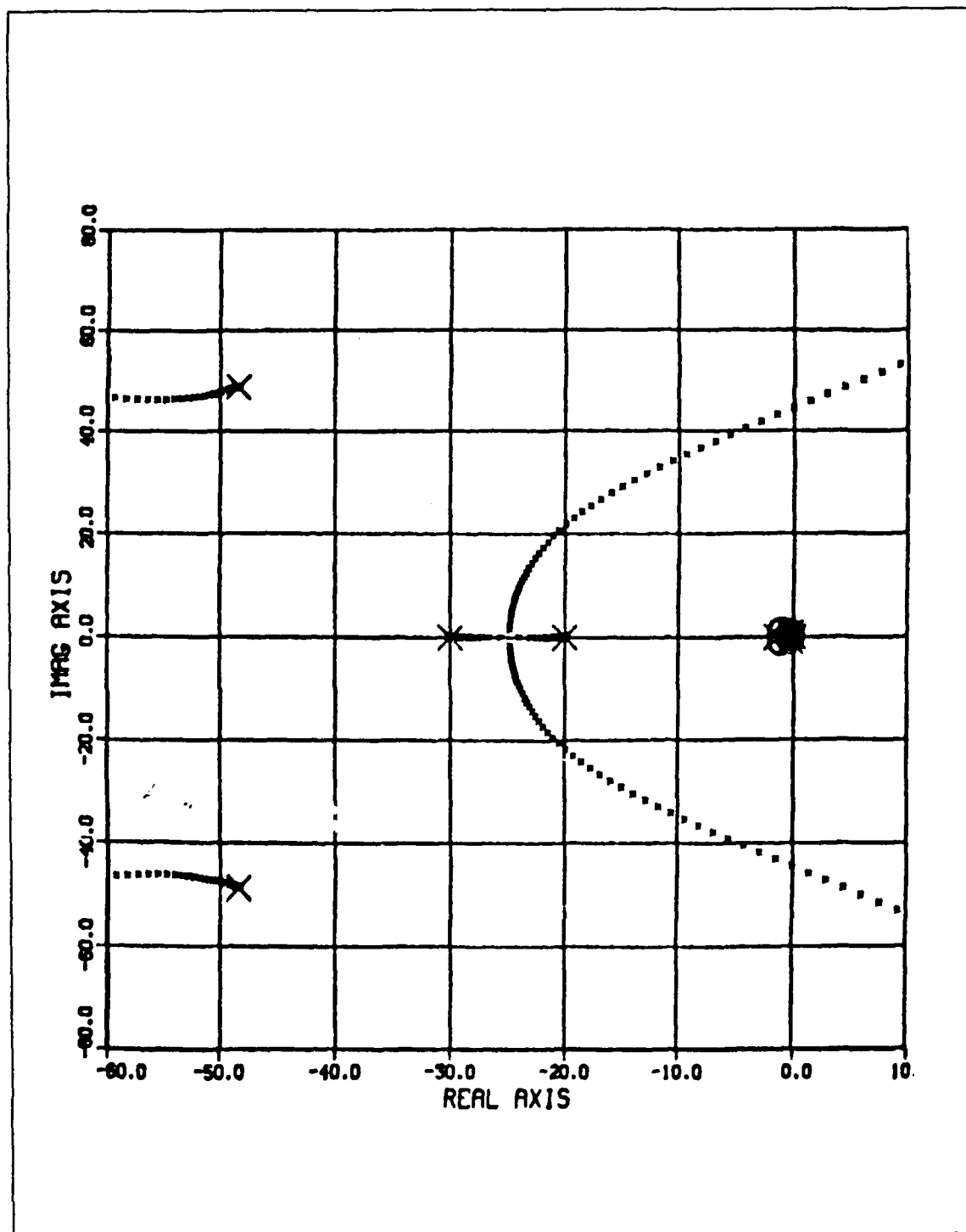


Figure 48. Root-Locus For Loop Transfer Function

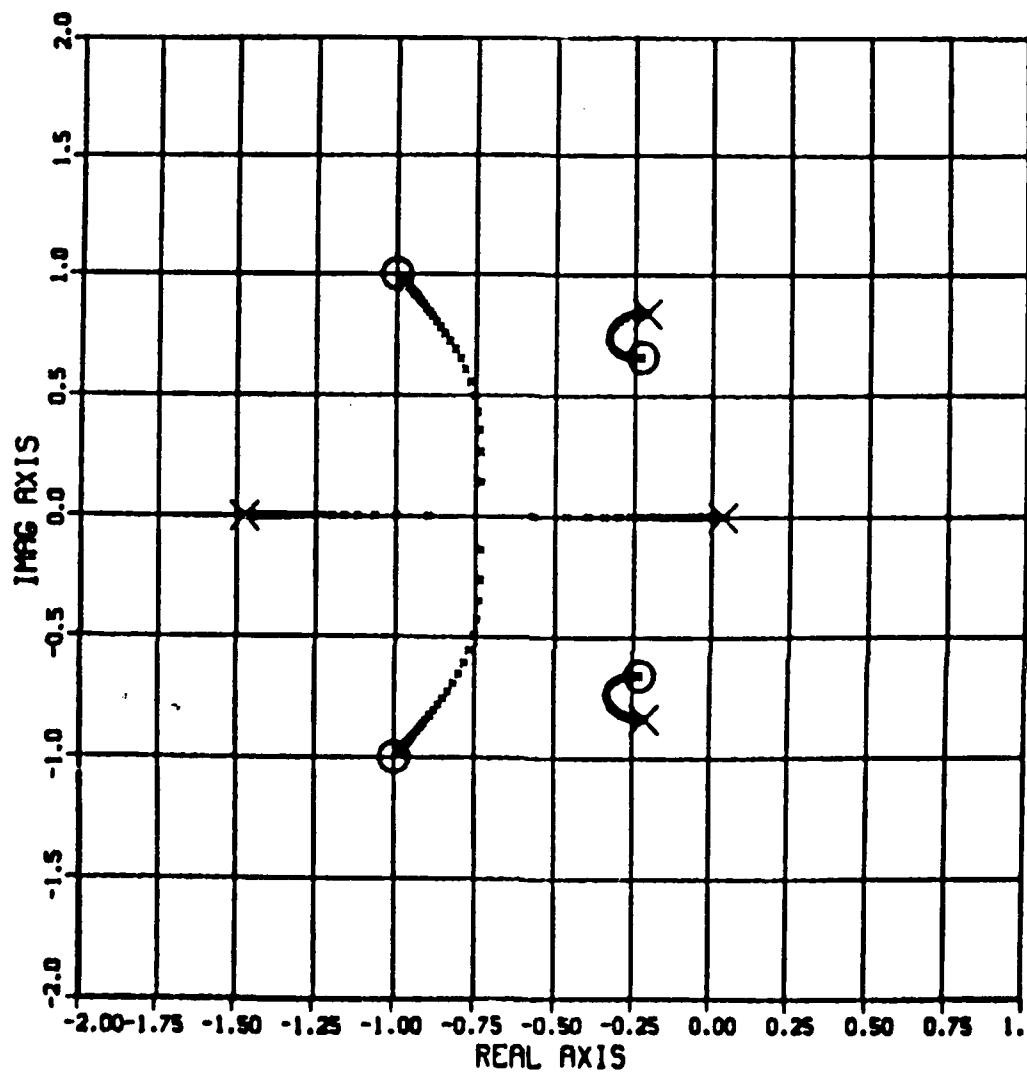


Figure 49. Magnified Root-Locus For Loop Transfer Function

To find a value for K_2 , we should get the tabulated points for the Root-Locus and see for which gain all of the roots are in the left half plane. For this particular filter the minimum value for K_2 was 2125.8. We selected $K_2 = 2500$ arbitrarily. For this value, our system becomes as in Figure 50, where

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (30)$$

$$H_2(s) = \frac{2500(s + 1 \mp j)}{(s + 20)(s + 30)} \quad (31)$$

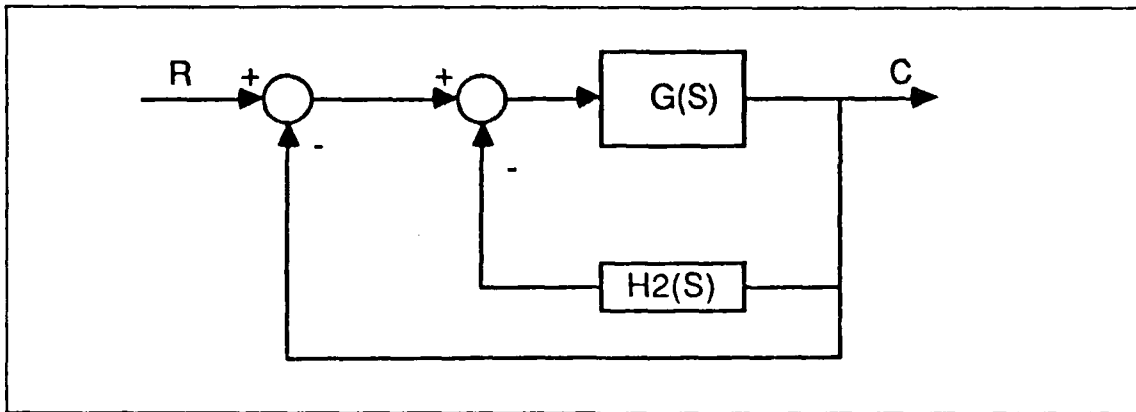


Figure 50. Compensated System

If we make the block diagram manipulation, we get the system shown in Figure 51, where

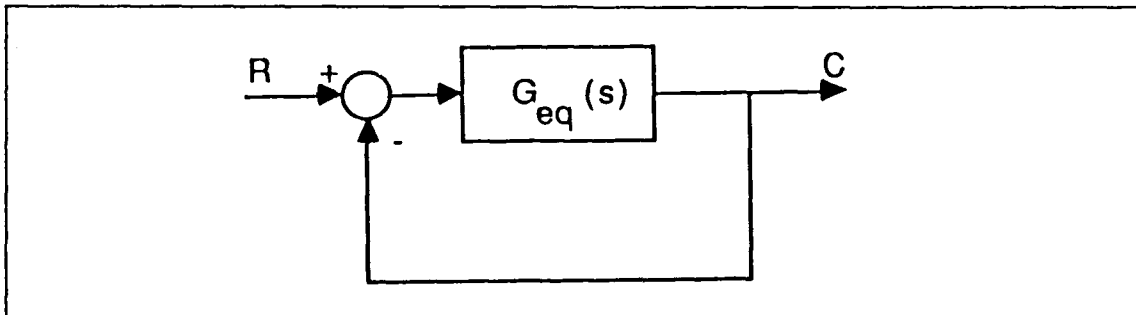


Figure 51. Resulting Equivalent G

$$G_{eq}(s) = \frac{77.187(s + 0.225 \mp j0.6617)(s + 20)(s + 30)}{(s + 0.007)(s + 0.22 \mp j0.84)(s + 1.427)(s + 24.5 \mp j6.3)(s + 48.87 \mp j48.3)} \quad (32)$$

The Root-Locus, magnified Root-Locus, open loop Bode plot and time response for the compensated system are given in Figures 52 - 55. As we may see from the Bode plot the system is stable but the gain is too low.

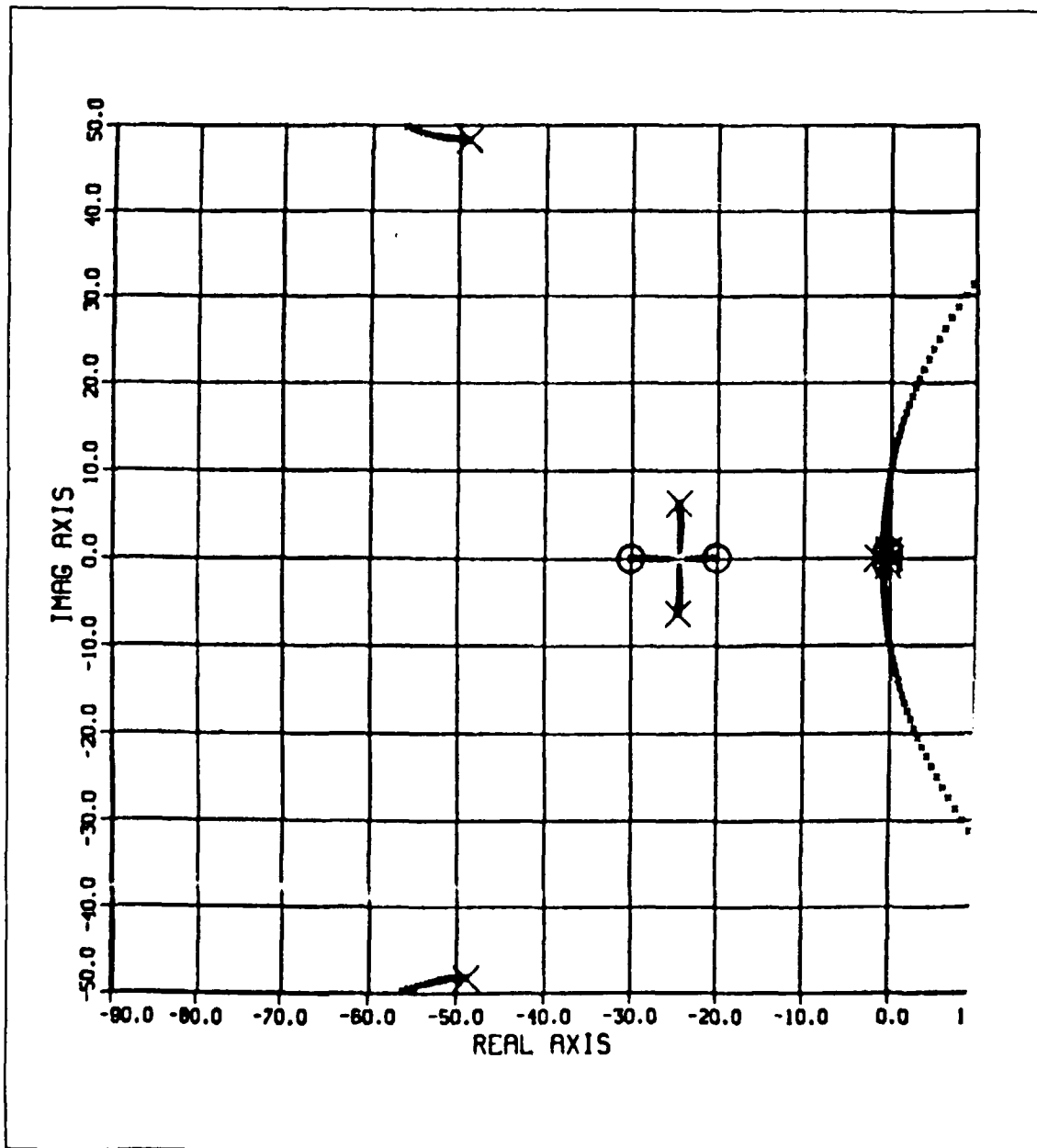


Figure 52. Root-Locus For Final System

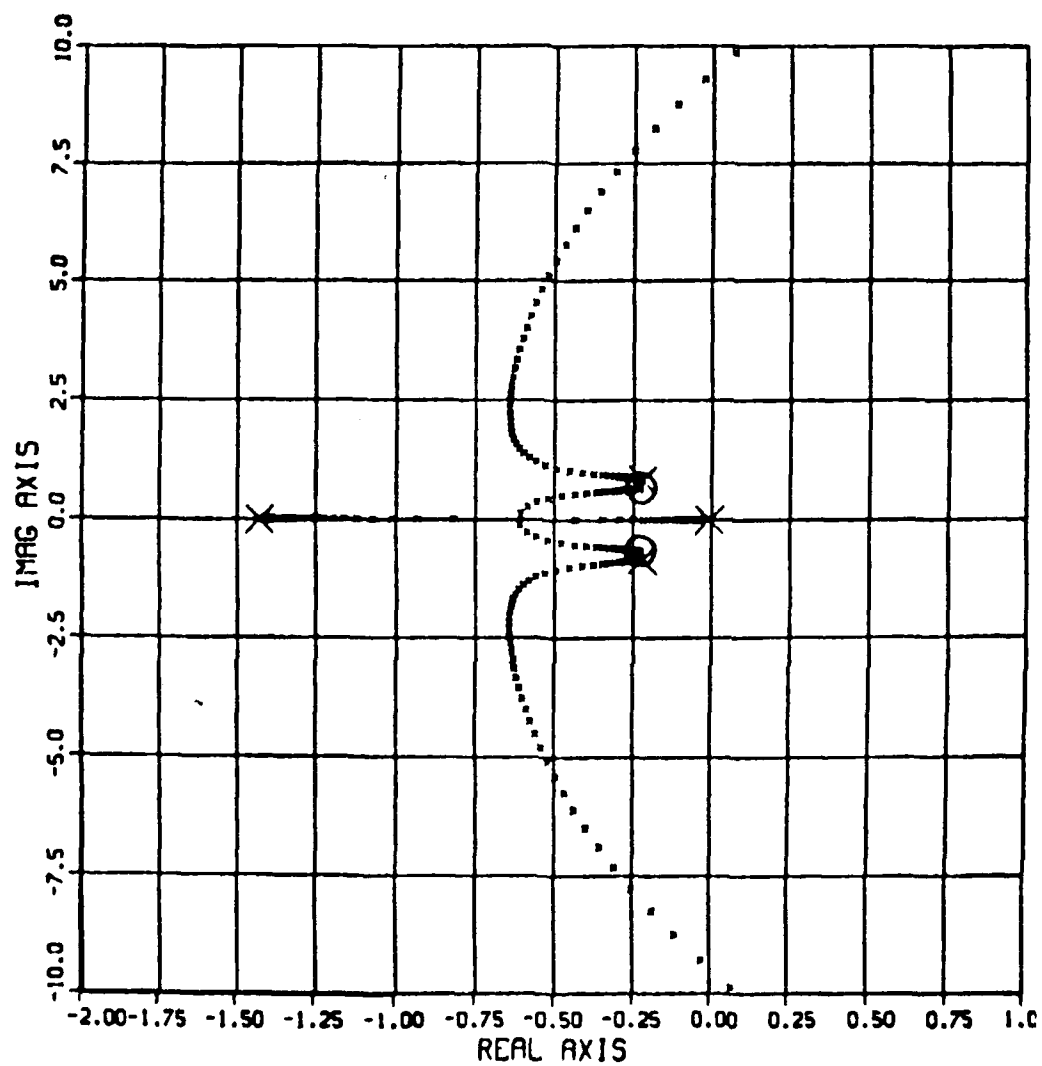


Figure 53. Magnified Root-Locus For Final System

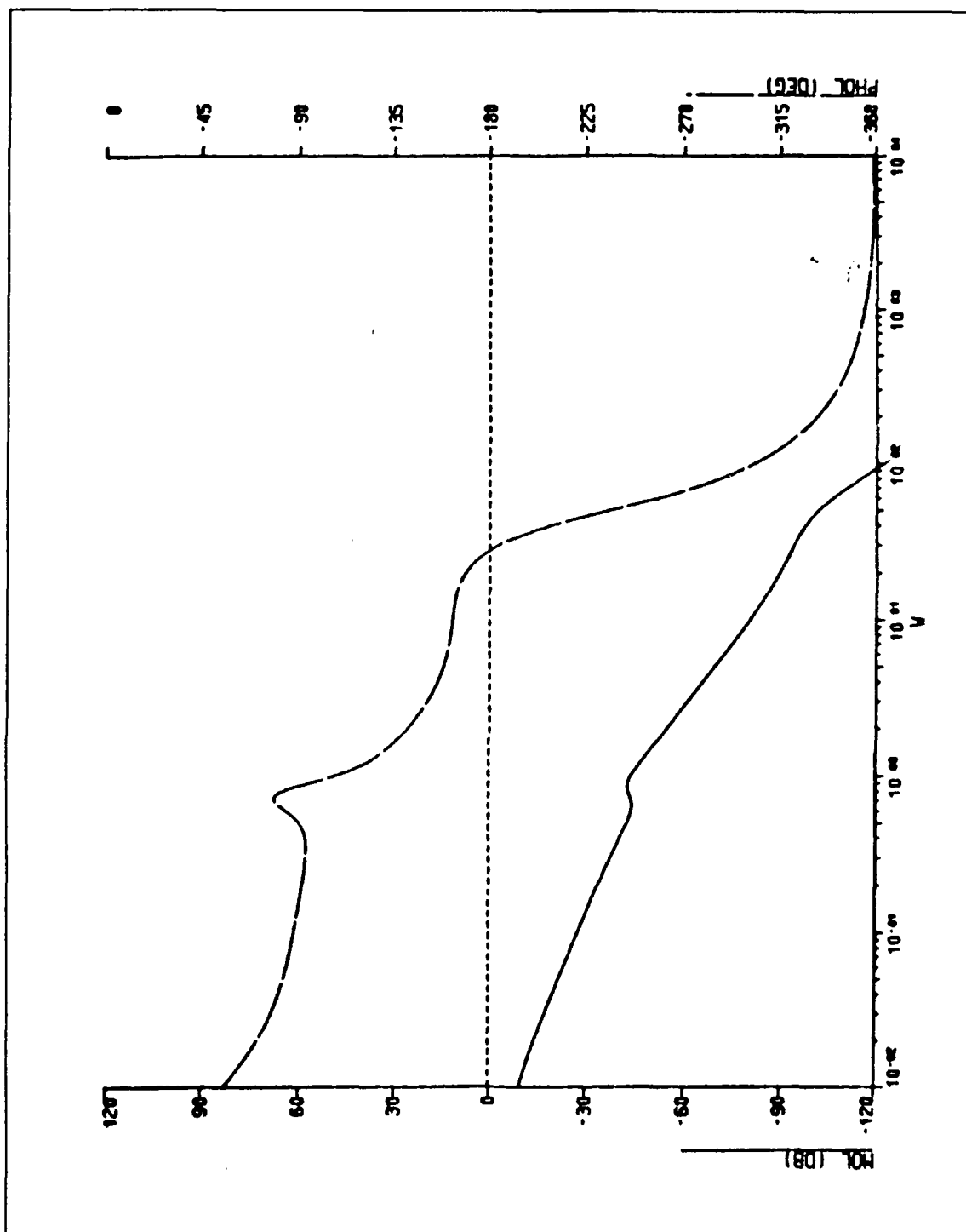


Figure 54. Bode Plot Without Outside Gain

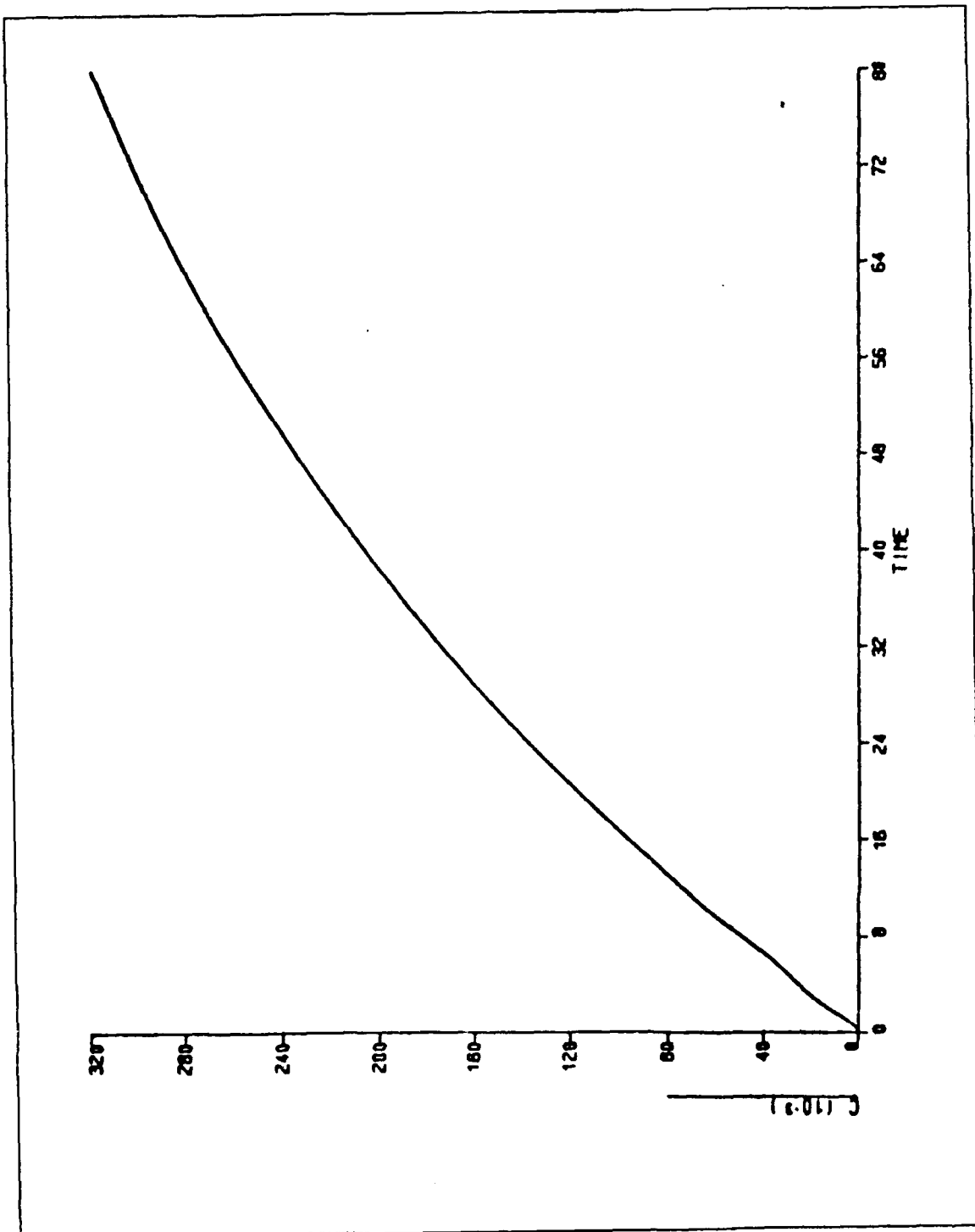


Figure 55. Time Response Without Outside Gain

We may handle this by putting another gain block outside the minor loop. The final system then becomes as in Figure 56.

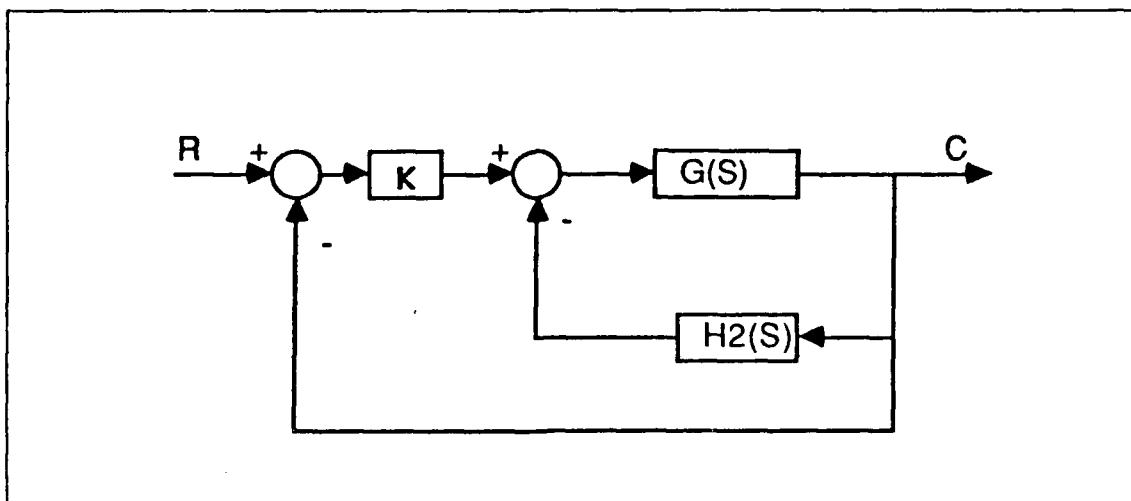


Figure 56. Final Compensated System

K may be selected according to the specifications. We chose $K = 100$ for illustration.

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (33)$$

$$H_2(s) = \frac{2500(s + 1 \mp j)}{(s + 20)(s + 30)} \quad (34)$$

The open loop Bode plot and time response for these particular values are given in Figures 57 and 58. We may change the loop gain according to the specifications. Effects of changing the loop gain will be shown with the next combination.

The third type of combination puts two complex zeros and one real pole close to the origin and one pole far away, all in the left half plane. In this case the loop transfer function becomes as in Figure 59, where

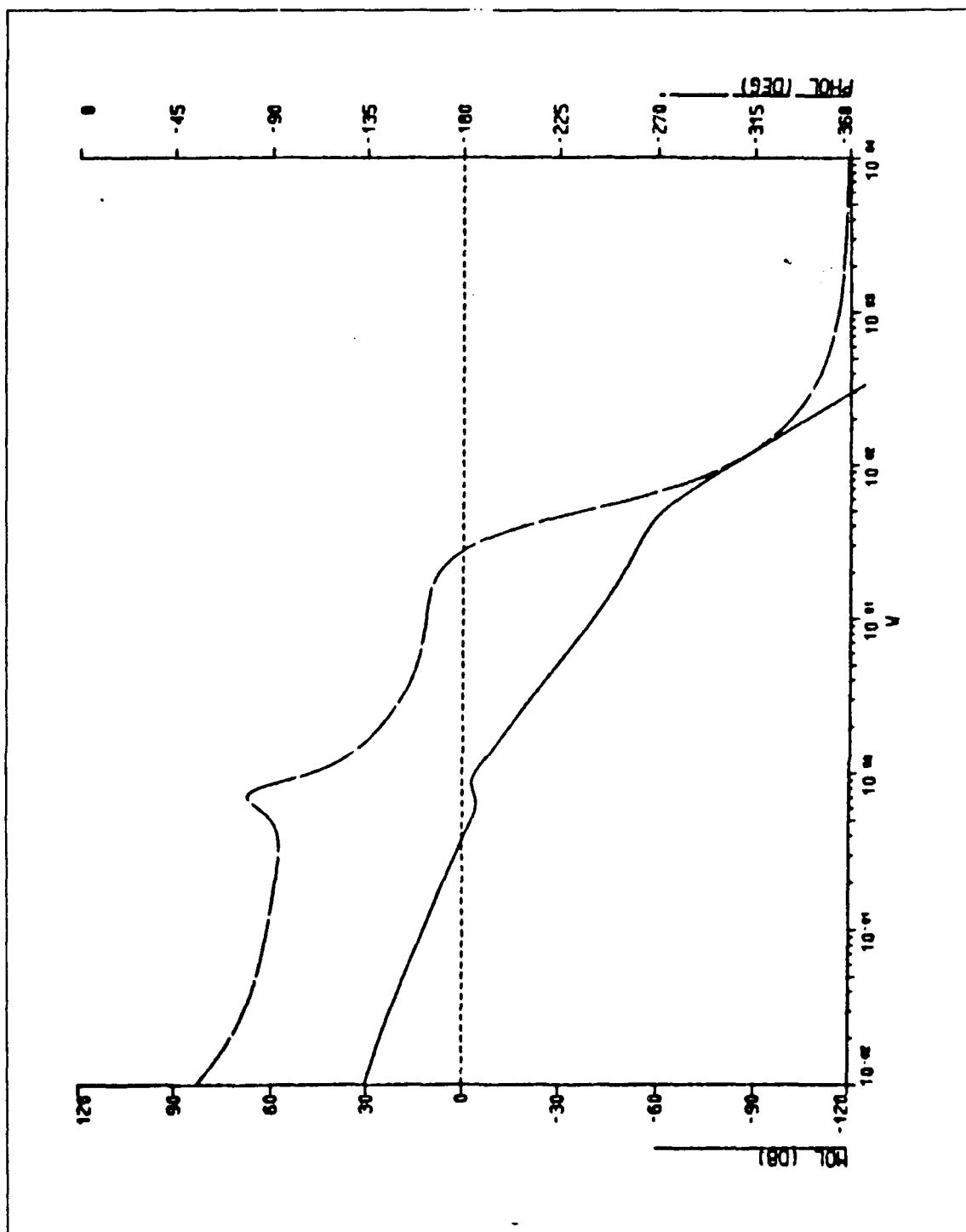


Figure 57. Bode Plot For Final System Loop Gain = 100

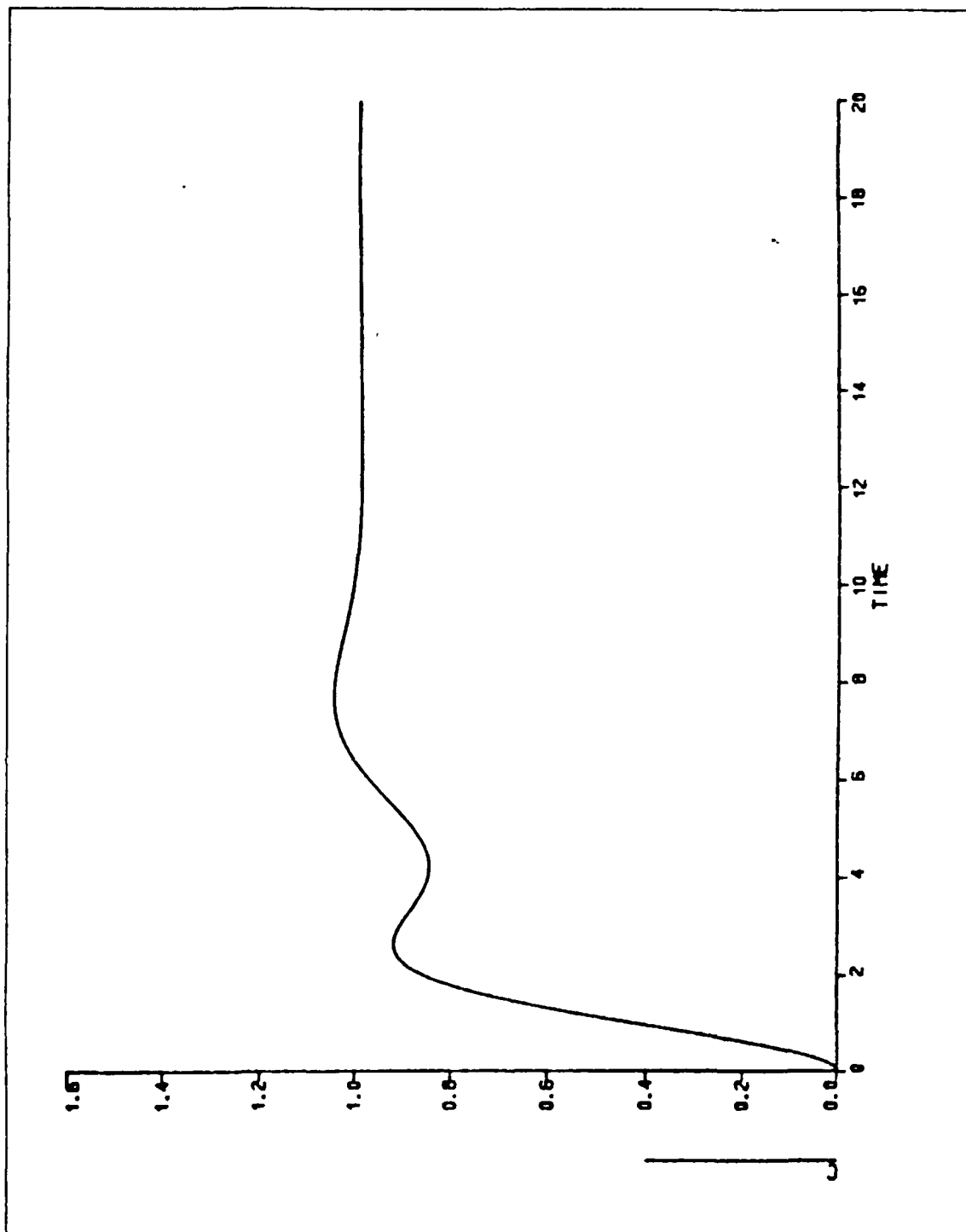


Figure 58. Final System Loop Gain = 100

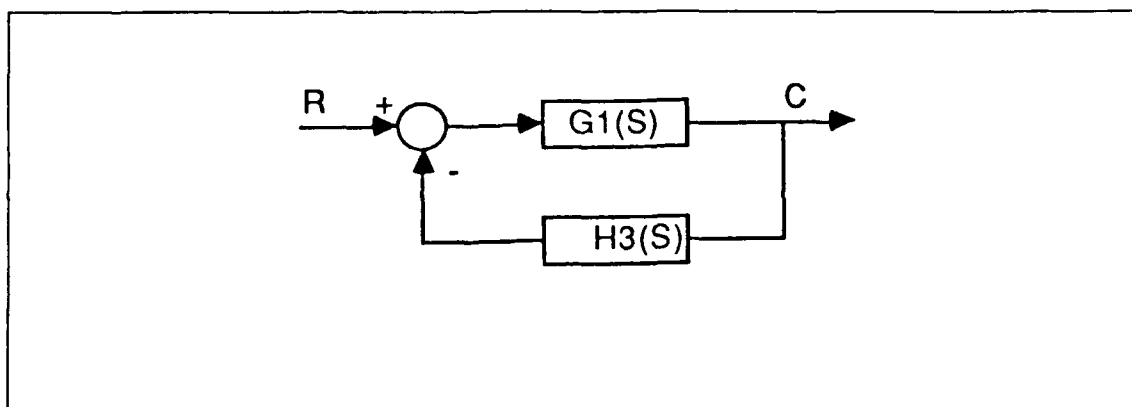


Figure 59. Loop Transfer Function Form

$$G_1(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.036)(s + 0.21 \mp j0.84)(s + 1.474)(s + 48.34 \mp j48.72)} \quad (35)$$

$$H_3(s) = \frac{K_3(s + 2 \mp j0.5)}{(s + 1)(s + 50)} \quad (36)$$

The Root-Locus and magnified Root-Locus for the loop transfer function are given in Figures 60 and 61. As we may see from the plots the locus is moved toward the left half plane.

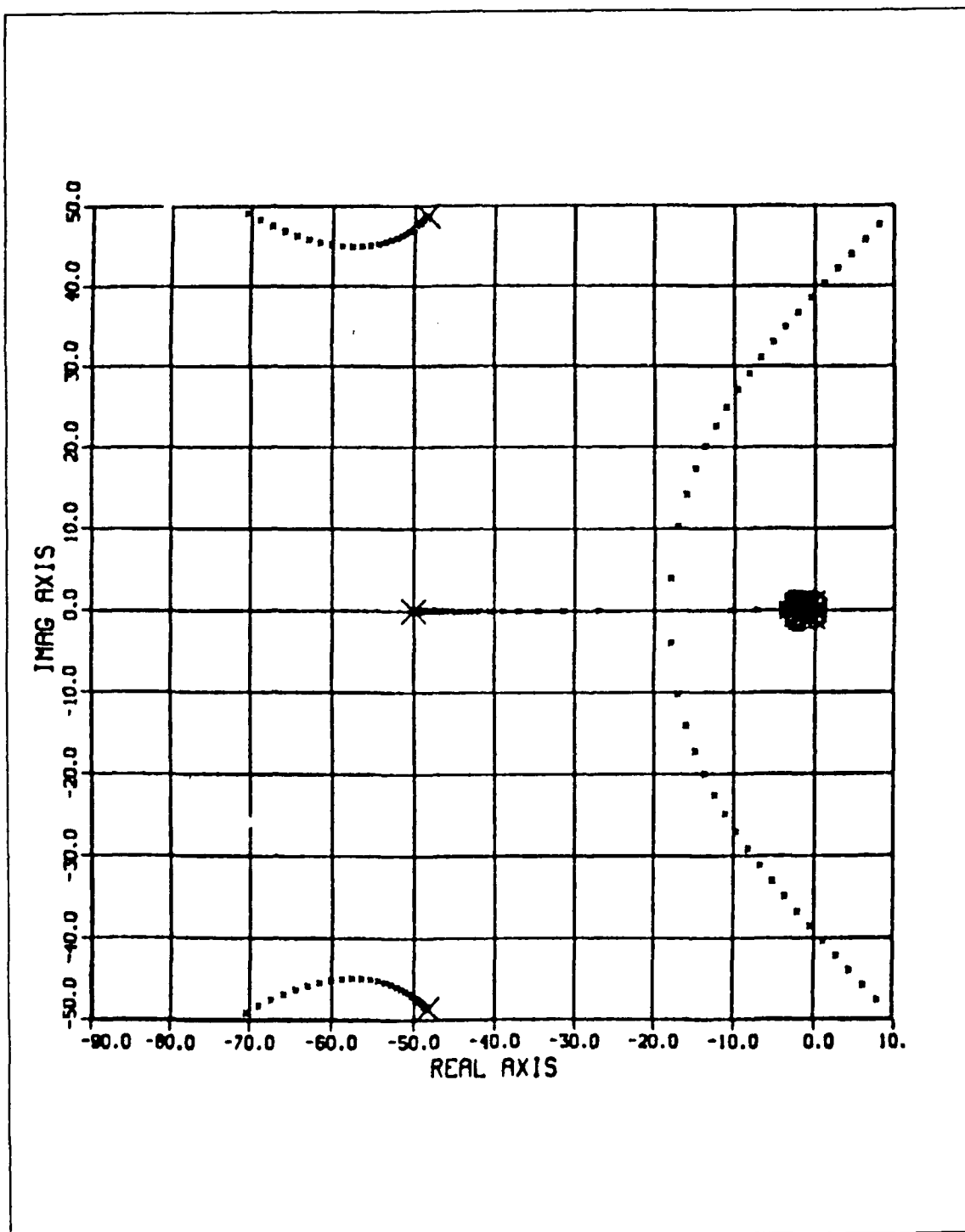


Figure 60. Root-Locus For Loop Transfer Function

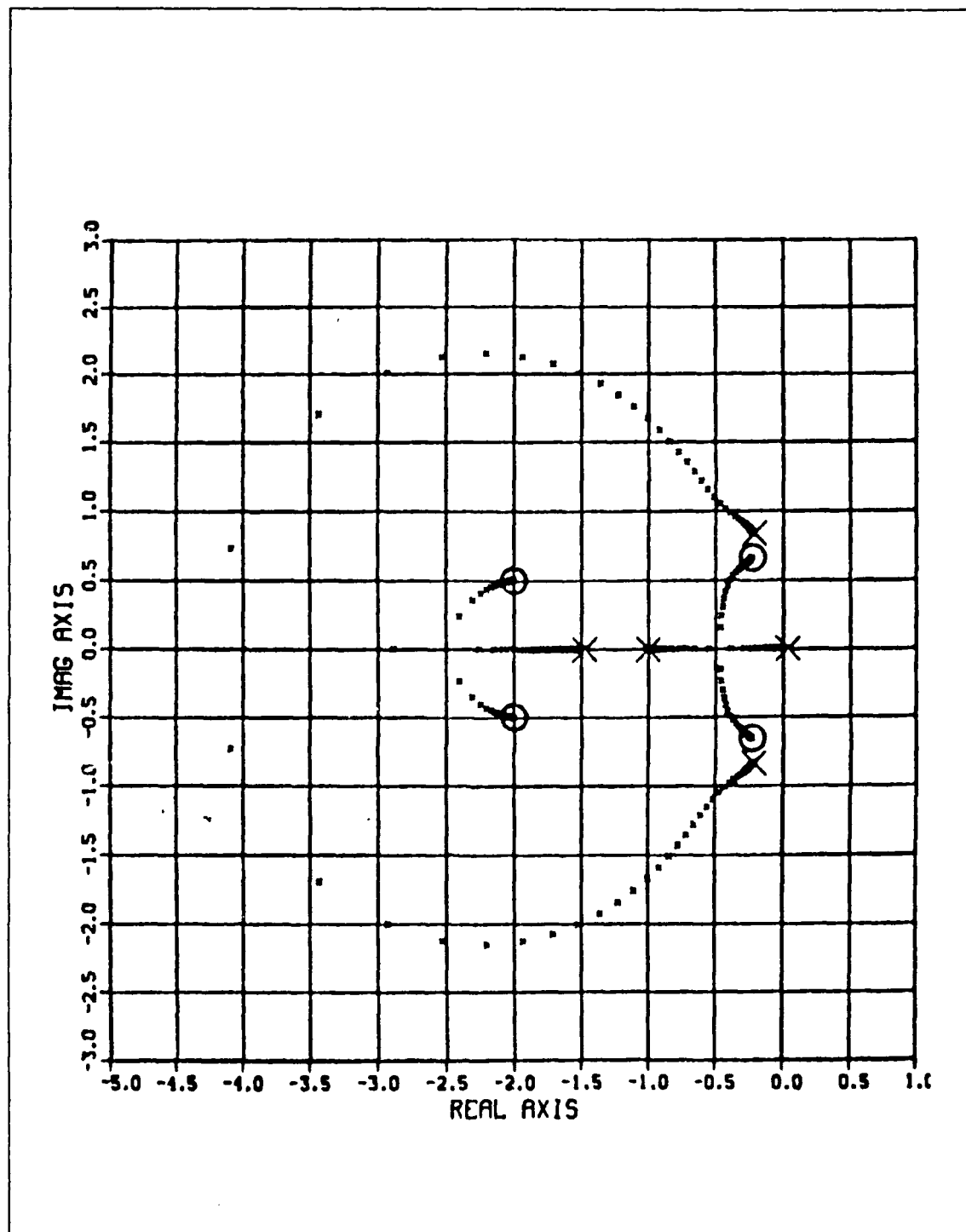


Figure 61. Magnified Root-Locus For Loop Transfer Function

To find a value for K_3 , we should get the tabulated points for the Root-Locus and see for which gain all of the roots are in the left half plane. For this particular filter the minimum value for K_3 was 81.6. We selected $K_3 = 250$ arbitrarily. By this, our system becomes as in Figure 62, where

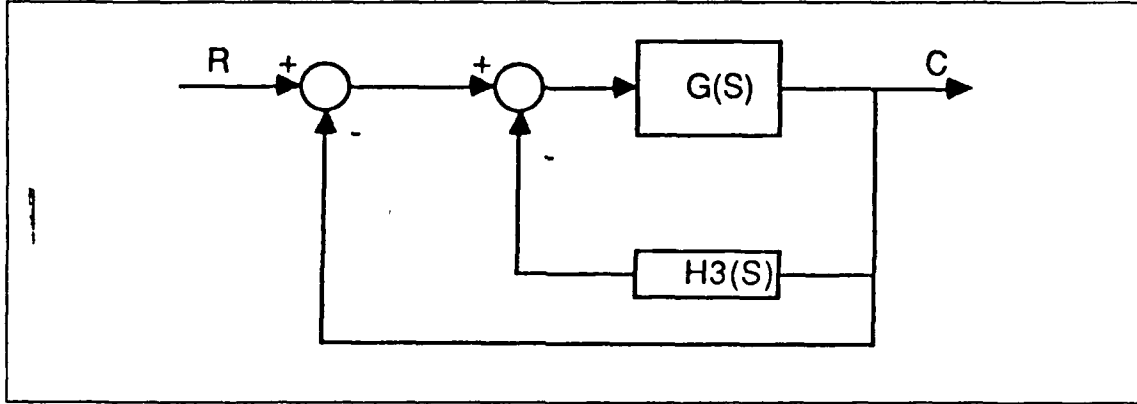


Figure 62. Compensated System

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (37)$$

$$H_3(s) = \frac{250(s + 2 \mp 0.5j)}{(s + 1)(s + 50)} \quad (38)$$

If we make the block diagram manipulation, we get the system shown in Figure 63, where

$$G_{eq}(s) = \frac{77.187(s + 0.225 \mp j0.6617)(s + 1)(s + 50)}{(s + 0.19)(s + 0.236 \mp j0.87)(s + 1.52)(s + 49.84)(s + 48.37 \mp j48.6)} \quad (39)$$

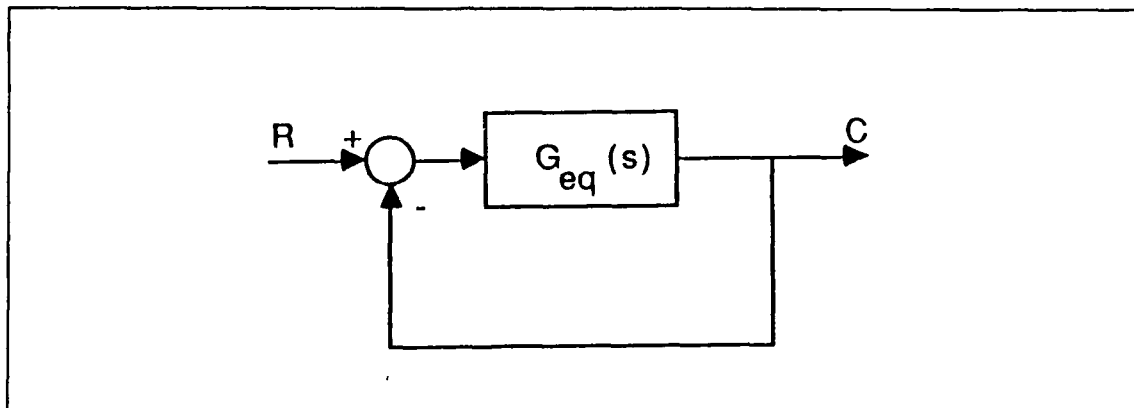


Figure 63. Resulting Equivalent G

The open loop Bode plot and time response for the compensated system are given in Figures 64 and 65. As we can see from the Bode plot the system is stable but the gain is too low.

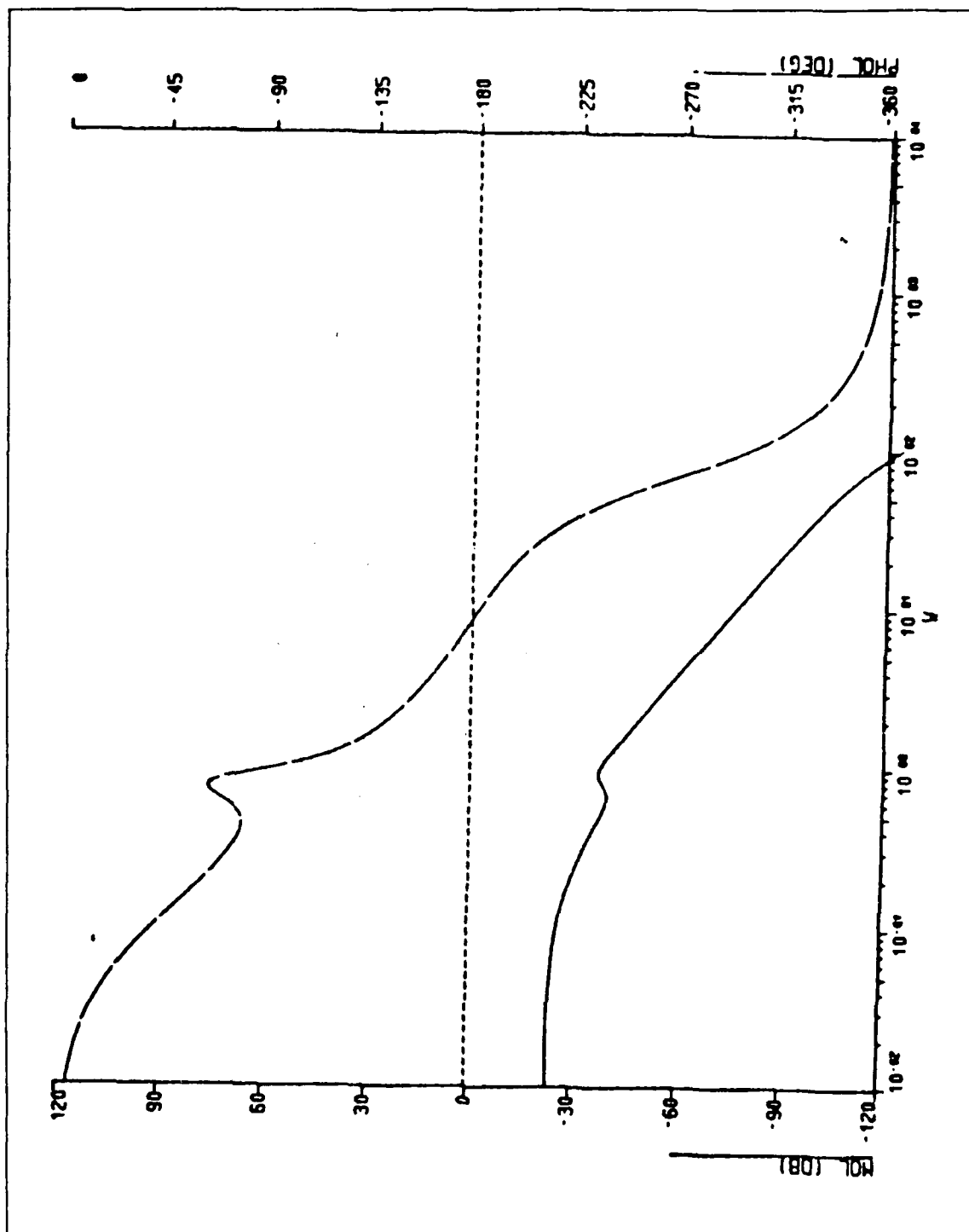


Figure 64. Bode Plot Without Outside Gain

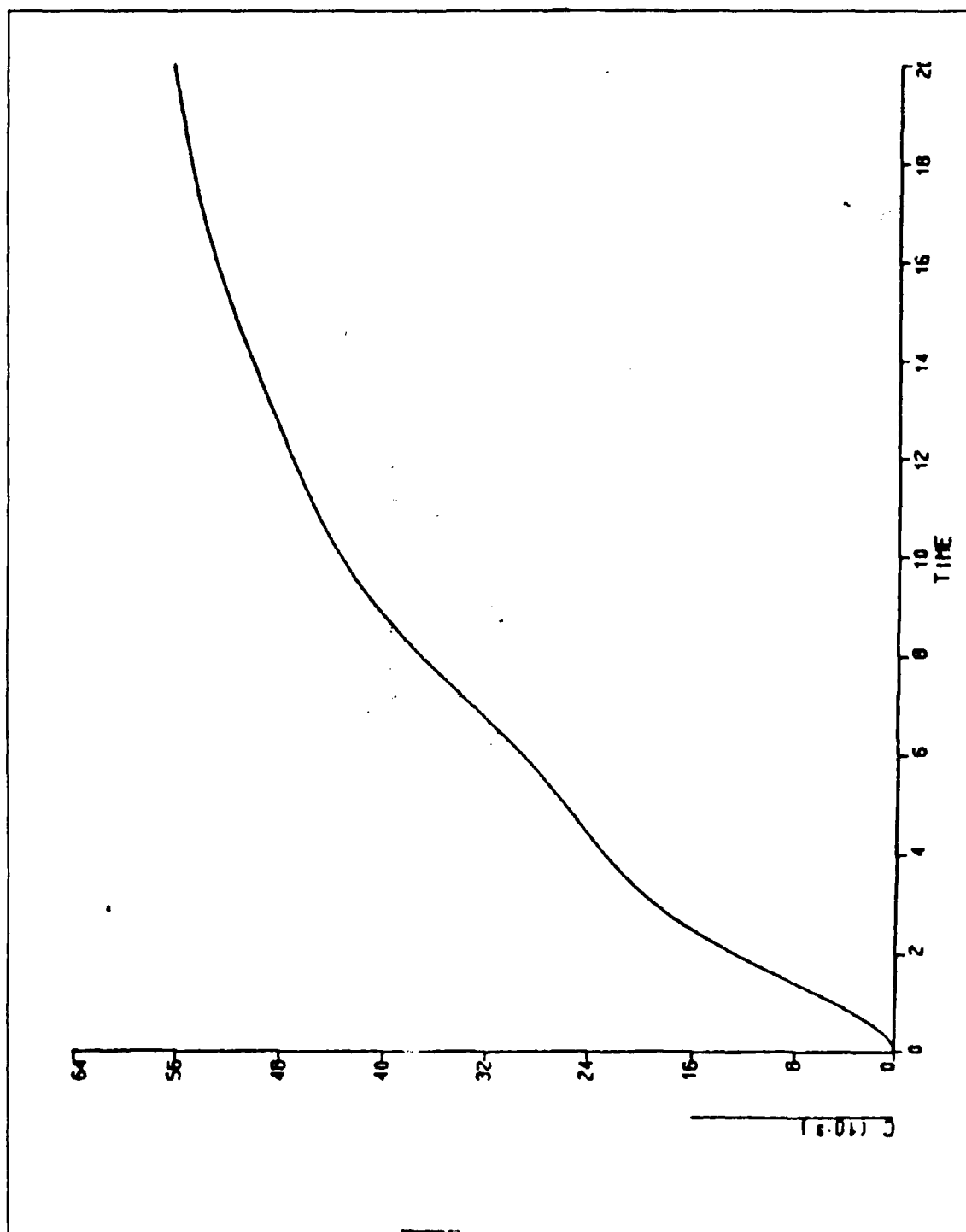


Figure 65. Time Response Without Outside Gain

We may handle this by putting another gain block outside the minor loop. The final system then becomes as in Figure 66.

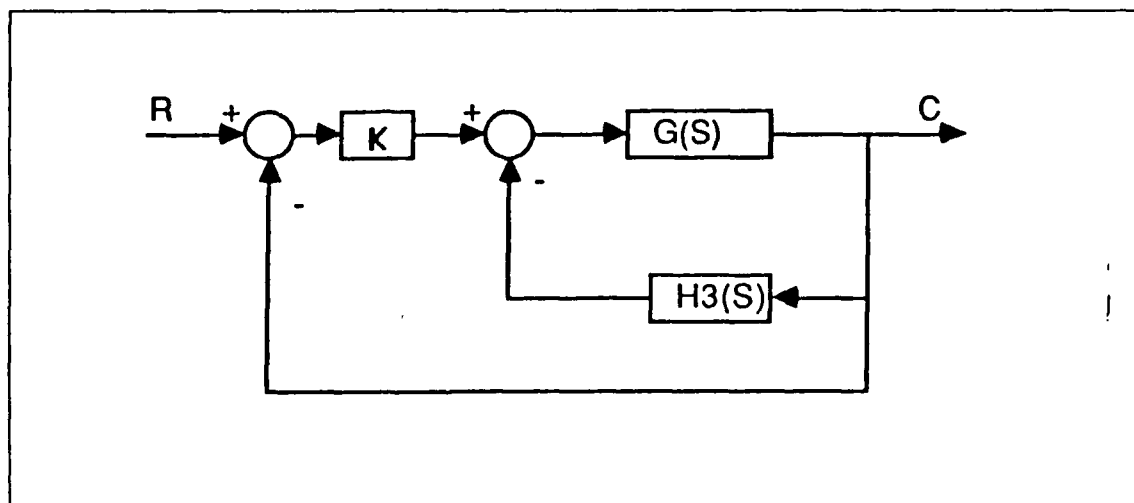


Figure 66. Final Compensated System

K may be selected according to the specifications. We chose different values for illustration.

$$G(s) = \frac{77.187(s + 0.225 \mp j0.6607)}{(s - 0.0434)(s + 1.464)(s + 0.21 \mp j0.844)(s + 48.35 \mp j49.34)} \quad (40)$$

$$H_3(s) = \frac{250(s + 2 \mp j0.5j)}{(s + 1)(s + 50)} \quad (41)$$

The open loop Bode plot and time response for $K = 100$ are given in Figures 67 and 68.

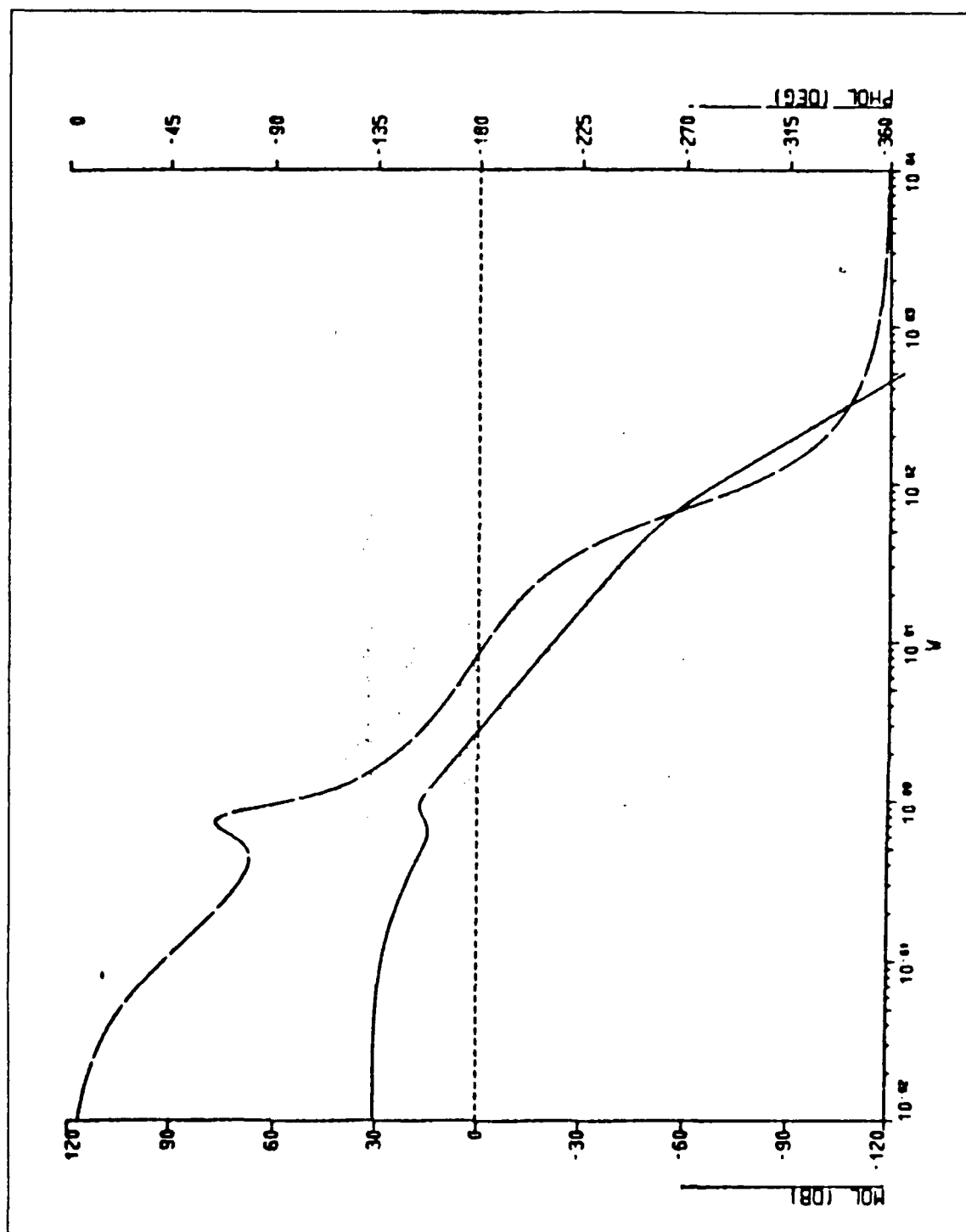


Figure 67. Bode Plot For Final System Loop Gain = 100

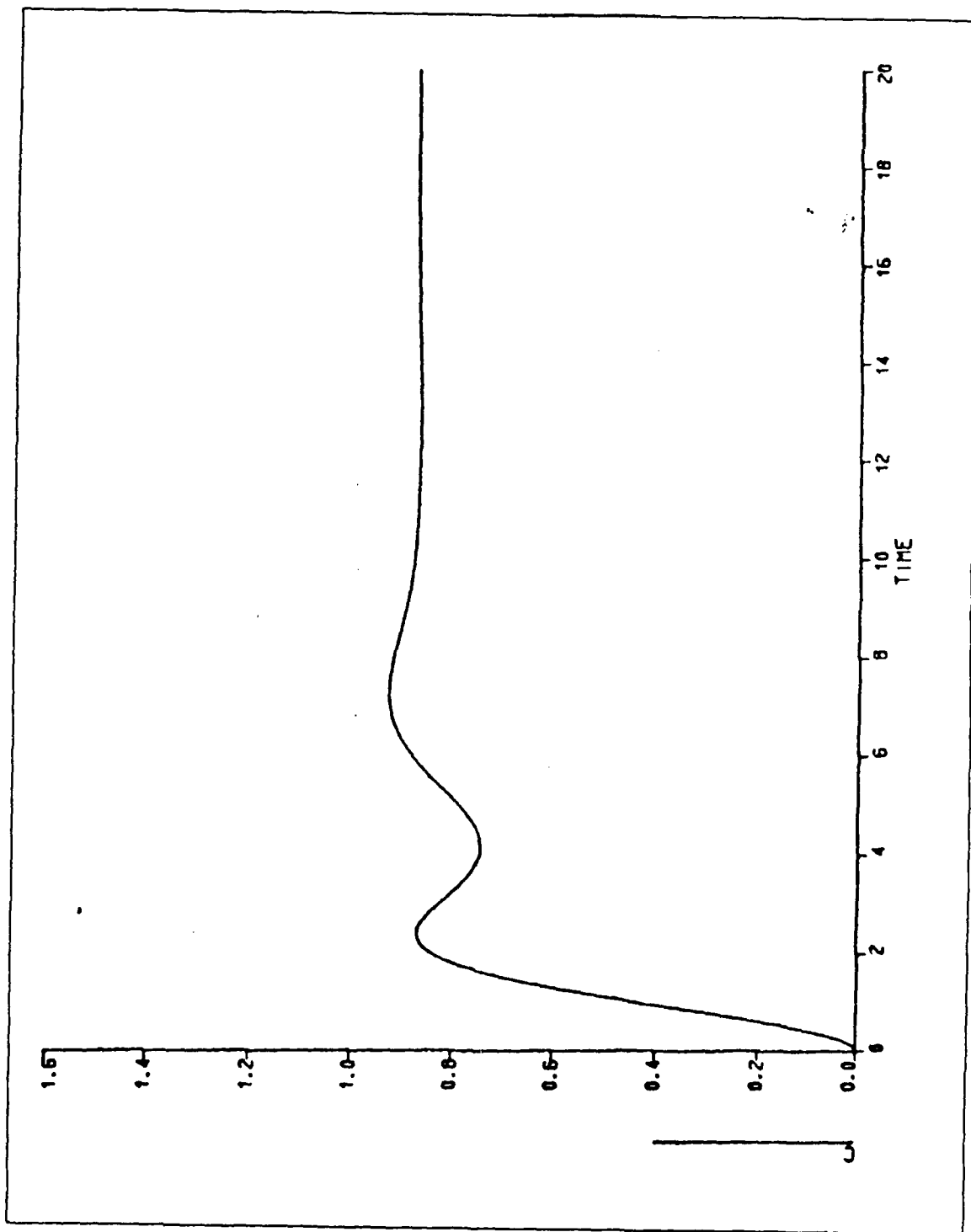


Figure 68. Final System Loop Gain = 100

We can change the loop gain according to the specifications. Figures 69 and 70 show the time responses for this system when this gain is 500 and 1000. As we may see from the time response we can change the amount of overshoot by changing this gain.

As a result of this problem we can summarize the steps for designing feedback compensators for non-minimum phase systems as following :

1. Find the closed loop transfer function and get the Root-Locus for this.
2. Pick the filter poles and zeros according to the Root-Locus rules.
3. Get the Root-Locus for the loop transfer function. Tabulate the gain values, and select the gain that puts all the roots to the left half plane.
4. Rearrange the system. Look for the Bode plot and time response. If they are acceptable, leave it that way otherwise select another gain outside the minor loop.

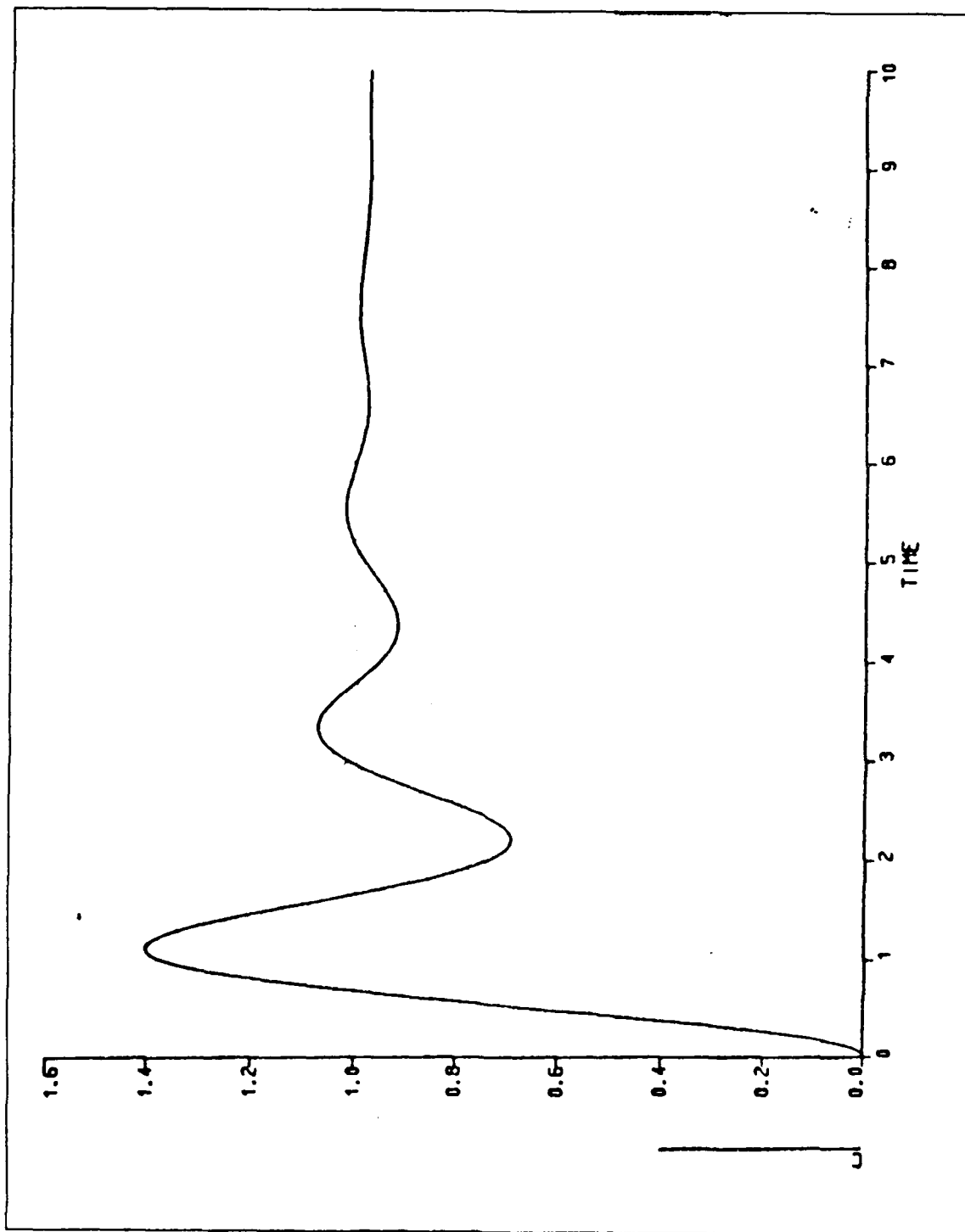


Figure 69. Final System Loop Gain = 500

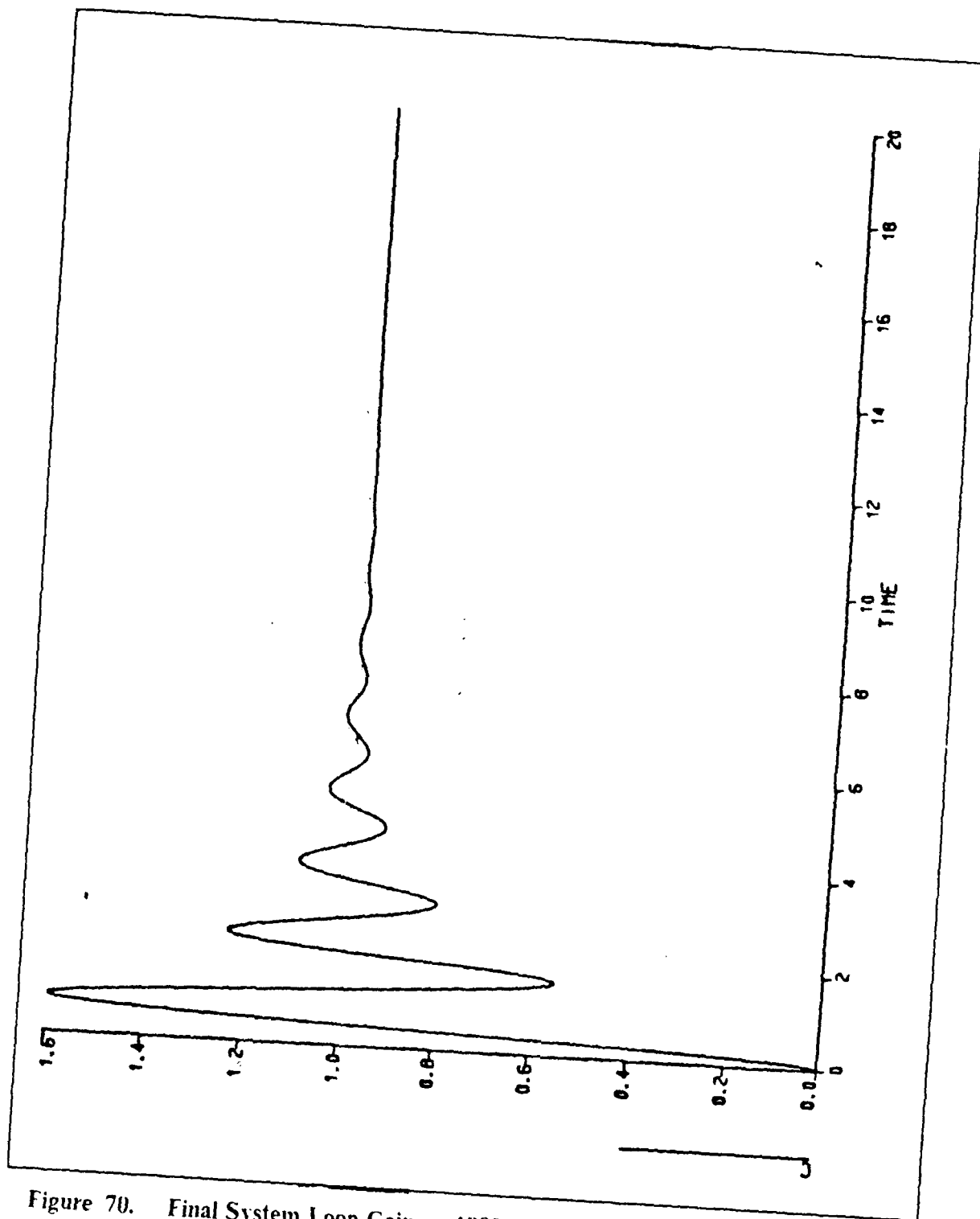


Figure 70. Final System Loop Gain = 1000

B. FLEXIBLE ROCKET CONTROL SYSTEM

As a second example we chose the control system of a flexible rocket which is shown in Figure 71. This example was extracted from Ref. 1.

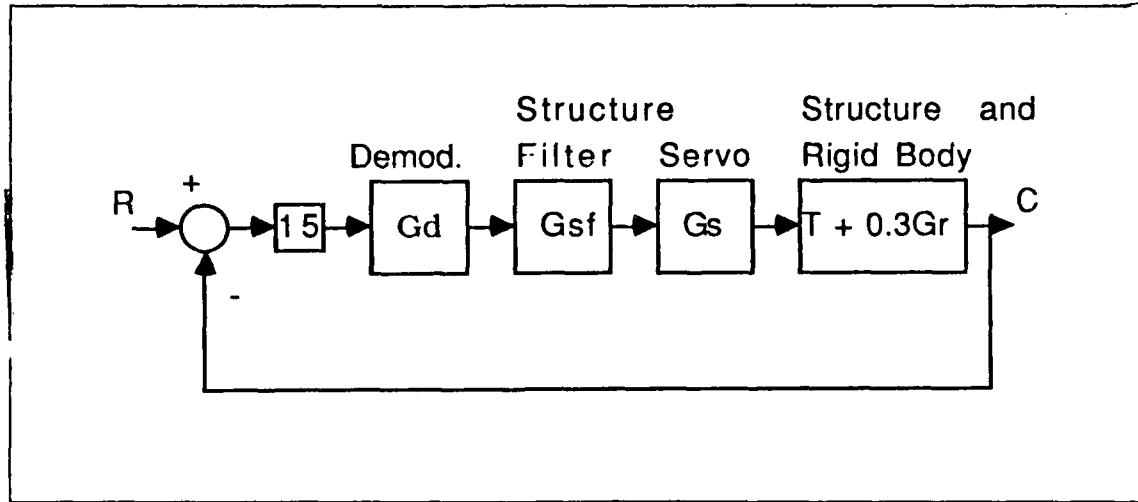


Figure 71. Original Uncompensated System

$G_R(s)$ is the transfer function for the rigid body and is defined as :

$$G_R(s) = \frac{7.21}{(s + 1.6)(s - 1.48)} \quad (42)$$

$$G_D(s) = \frac{333}{(s + 333)} \quad (43)$$

$$G_S(s) = \frac{2750}{s^2 + 42.2s + 2750} \quad (44)$$

$$G_{SF}(s) = \frac{(s^2 + 70s + 4000)(s^2 + 22s + 12800)}{(s^2 + 30s + 5810)(s^2 + 30s + 12800)} \quad (45)$$

And the structure is defined as ;

$$T(s) = \frac{0.686(s + 53)(s - 53)(s^2 - 152.2s + 14500)(s^2 + 153.8s + 14500)}{(s^2 + s + 605)(s^2 + 45.5s + 2660)(s^2 + 2.51s + 3900)(s^2 + 3.99s + 22980)} \quad (46)$$

After performing all calculations the total system becomes as in Figure 72. Since the order of the numerator and denominator are quite large, the transfer function is split into three parts to fit on the page.

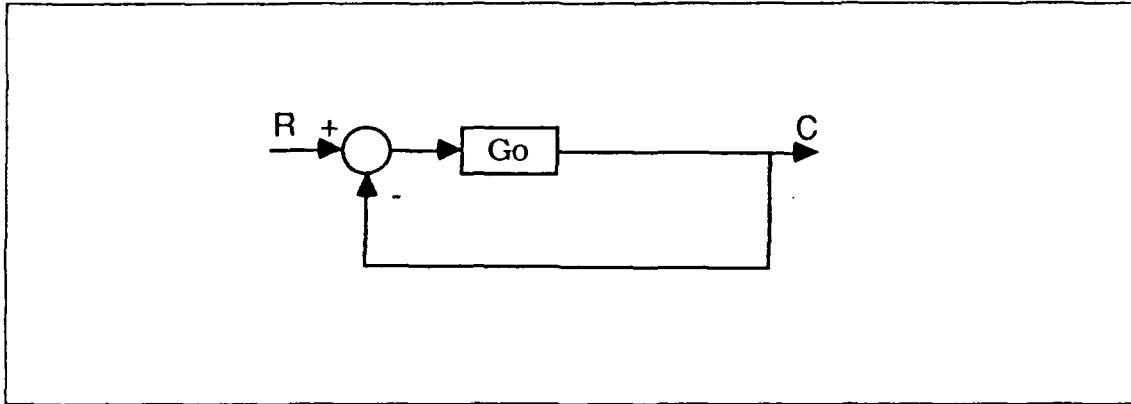


Figure 72. Combined Uncompensated System

$G_o(s) = G_{o1} \times G_{o2} \times G_{o3}$ where ;

$$G_{o1} = \frac{39134576.25(s + 35 \mp j52.7)(s + 11 \mp j112.6)}{(s + 15 \mp j74.7)(s + 15 \mp j112.1)(s + 333)(s + 21.1 \mp j48)} \quad (47)$$

$$G_{o2} = \frac{(s - 21.55 \mp j21.79)(s + 72.76 \mp j100.24)(s - 83.1 \mp j61.82)}{(s + 22.75 \mp j46.29)(s + 1.255 \mp j62.44)(s + 1.995 \mp j151.57)(s - 1.48)} \quad (48)$$

$$G_{o3} = \frac{(s + 11.6 \mp j24.02)}{(s + 1.6)(s + 0.5 \mp j24.6)} \quad (49)$$

As we may see from the transfer function we have four zeros and a real pole in the right half plane. In addition to this most of the poles and zeros which are in the left half plane are very close to the imaginary axis.

The Root-Locus for the open loop uncompensated system is given by Figure 73. The system is unstable. Since there is a real pole in the right half plane we can not easily interpret the Bode plot for this system.

The next step is to find the loop transfer function as in previous examples. By performing the regular block diagram manipulation the closed loop transfer function becomes $G_c(s) = G_{o1}(s) \times G_{o2}(s) \times G_{o3}(s)$ which is

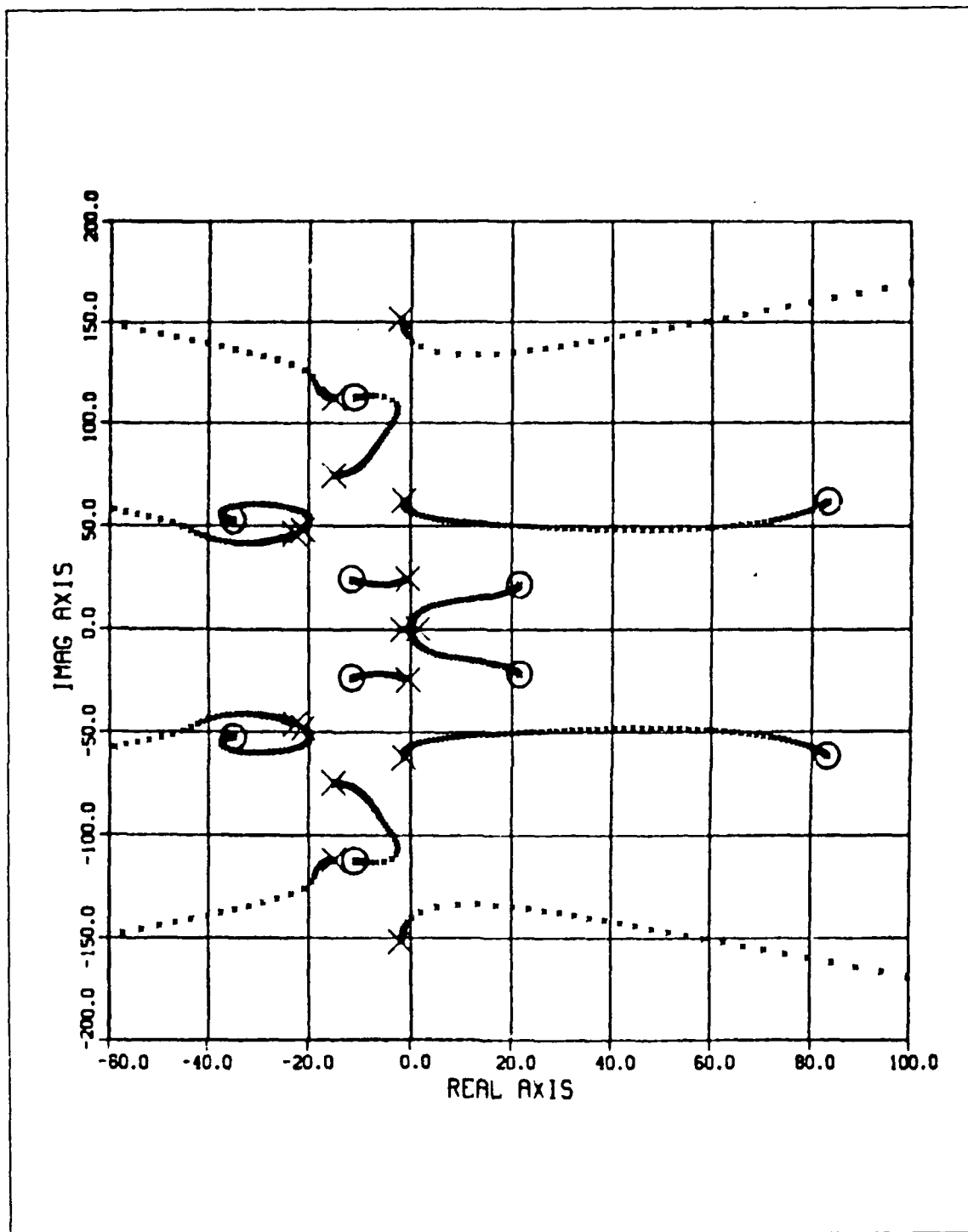


Figure 73. Root-Locus For Uncompensated System

$$G_{c1}(s) = \frac{39134576.25(s + 35 \mp j52.7)(s + 11 \mp j112.6)(s + 11.6 \mp j24)}{(s + 333)(s + 1.9 \mp j151.5)(s + 15 \mp j112.2)(s + 13.8 \mp j74.5)} \quad (50)$$

$$G_{c2}(s) = \frac{(s + 72.8 \mp j100.2)(s - 83 \mp j61.8)(s - 21.6 \mp j21.8)}{(s + 1.5 \mp j60.4)(s + 19.9 \mp j52.6)(s + 1 \mp j24)(s - 0.5 \mp j4.5)} \quad (51)$$

$$G_{c3}(s) = \frac{1}{(s + 24.8 \mp j44.17)} \quad (52)$$

From this point on we may start to think in terms of Root-Locus design. As we saw from the loop transfer function we have a pair of complex poles and four zeros in the right half plane. In addition to this most of the poles in the left half plane are very close to imaginary axis and there are four zeros in the right half plane. So probably some of the roots are going to end up at these zeros. This will give us a gain constraint.

Since we have poles close to the origin and imaginary axis in the left half plane, putting a zero or zeros at the origin would not help us, because in this case we always will have roots in the right half plane. So we decided to use a general type of compensator which is two zeros close to the origin and two poles far away. The purpose of the zeros was to have loci from the poles in the right half plane terminate on these zeros. We made several trials with this type configuration but none of them worked. An illustration of this is shown in Figure 74, where $G_c(s)$ is the closed loop transfer function and

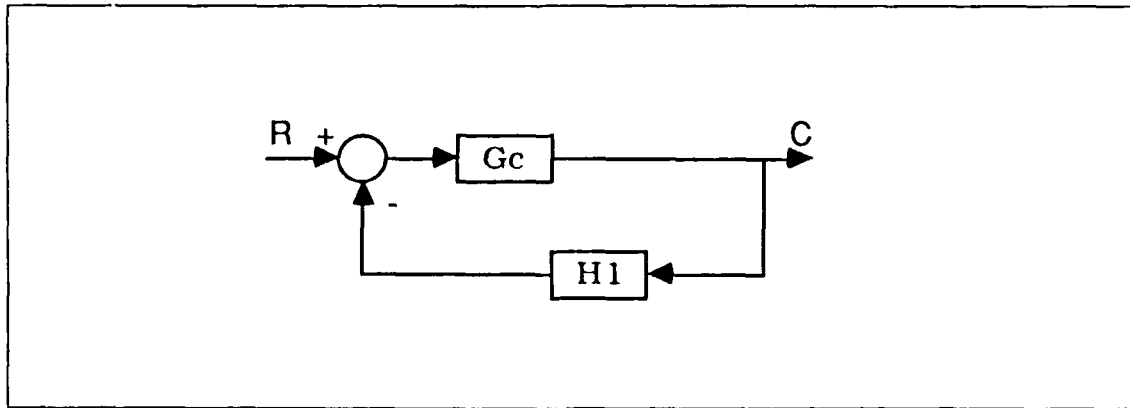


Figure 74. Illustration of General Configuration

$$H_1(s) = \frac{98.6(s + 0.46 \mp j4.482)}{(s + 100)(s + 200)} \quad (53)$$

The Root-Locus for this case is given in Figure 75.

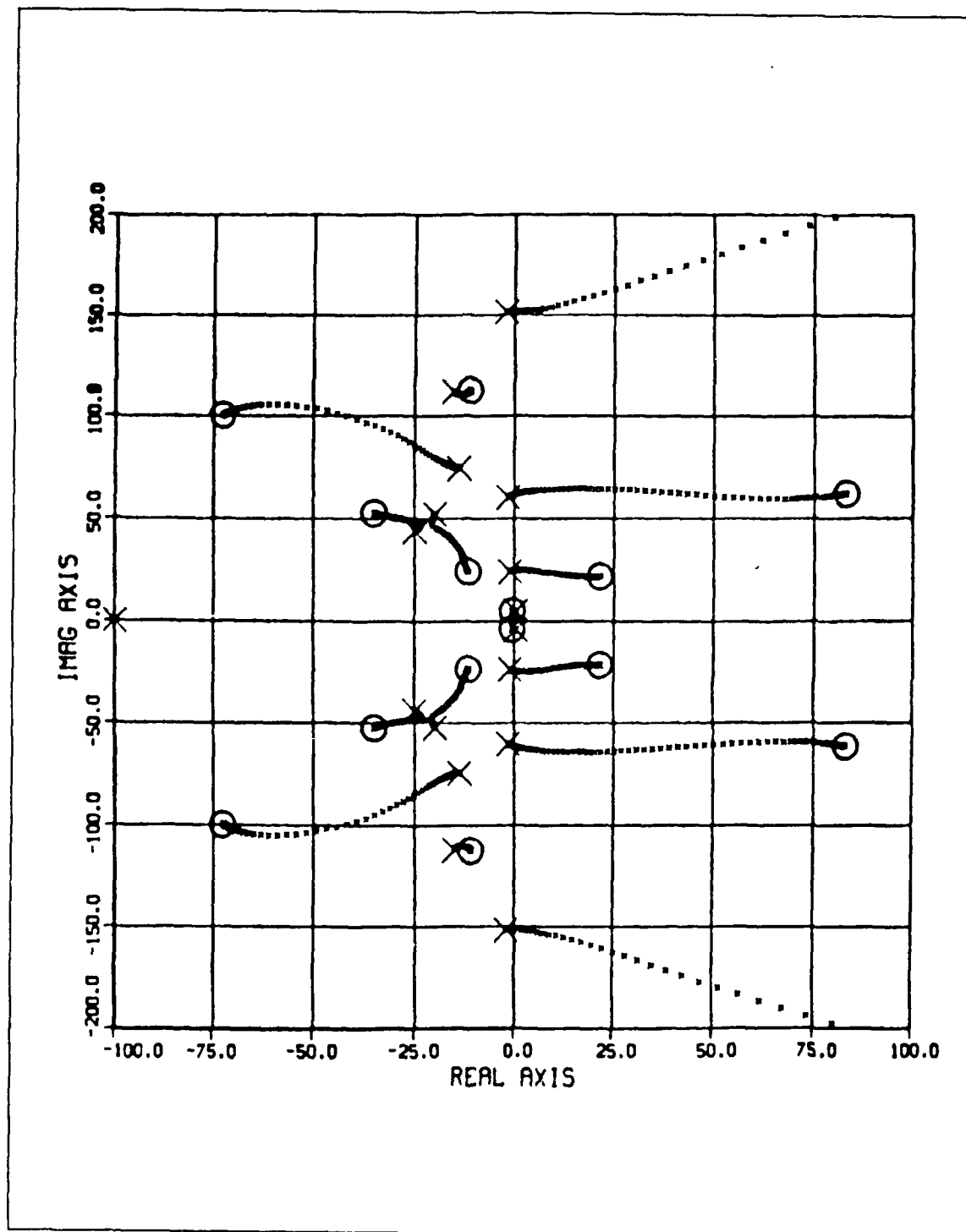


Figure 75. Root-Locus for General Type Compensated System

As we may see the system is unstable, because roots in the left half plane were moving faster than the roots in the right half plane.

With this type of problem, since the system has a 12th order numerator and 17th order denominator, designing the compensator by just examining the system would be impossible or require many trials. So we decided to use the function minimization subroutine in the DSL package in the mainframe. The parameters used in the subroutine and the function minimization program are given in Appendix A and B.

As the result of the program we found that we have to put the poles of the compensator close to the origin and to the left of the zeros which belong to the compensator. By putting these zeros and poles in this combination, we make the roots in the right half plane move faster. Figure 76 shows the block diagram for the compensated system, where $G_c(s)$ is the closed loop transfer function and

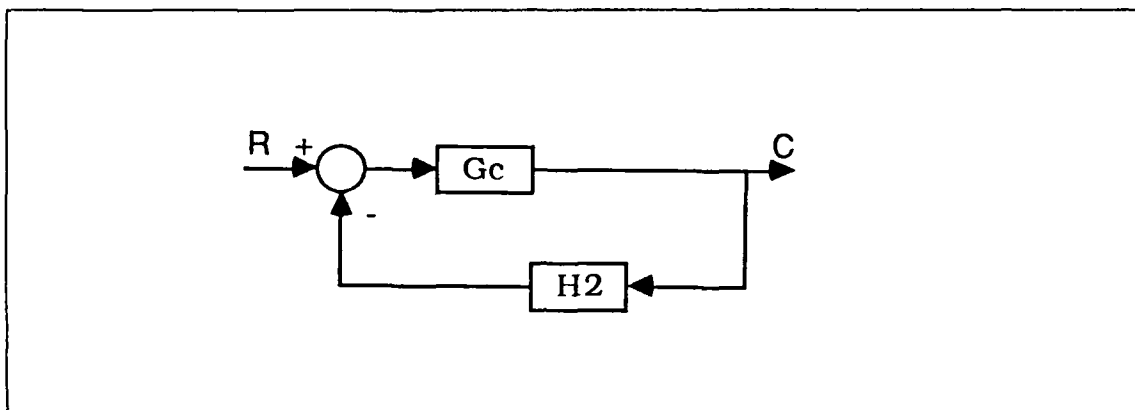


Figure 76. First Resulting Compensator of Function Minimization

$$H_2(s) = \frac{14.72(s + 0.875 \mp j2.058)}{(s + 2 \mp j8.343)} \quad (54)$$

The Root-Locus for this compensation is given in Figure 77. By getting the tabulated root locations, we see that for a Root-Locus gain of 0.068 we get all roots in the left half plane. Then our final compensator becomes ;

$$H_2(s) = \frac{s^2 + 1.75s + 5}{s^2 + 4s + 73.6} \quad (55)$$

The time response for this case is given in Figure 78. As we may see the system is stable but the settling time is a little long. According to the specs we may want a faster system than this.

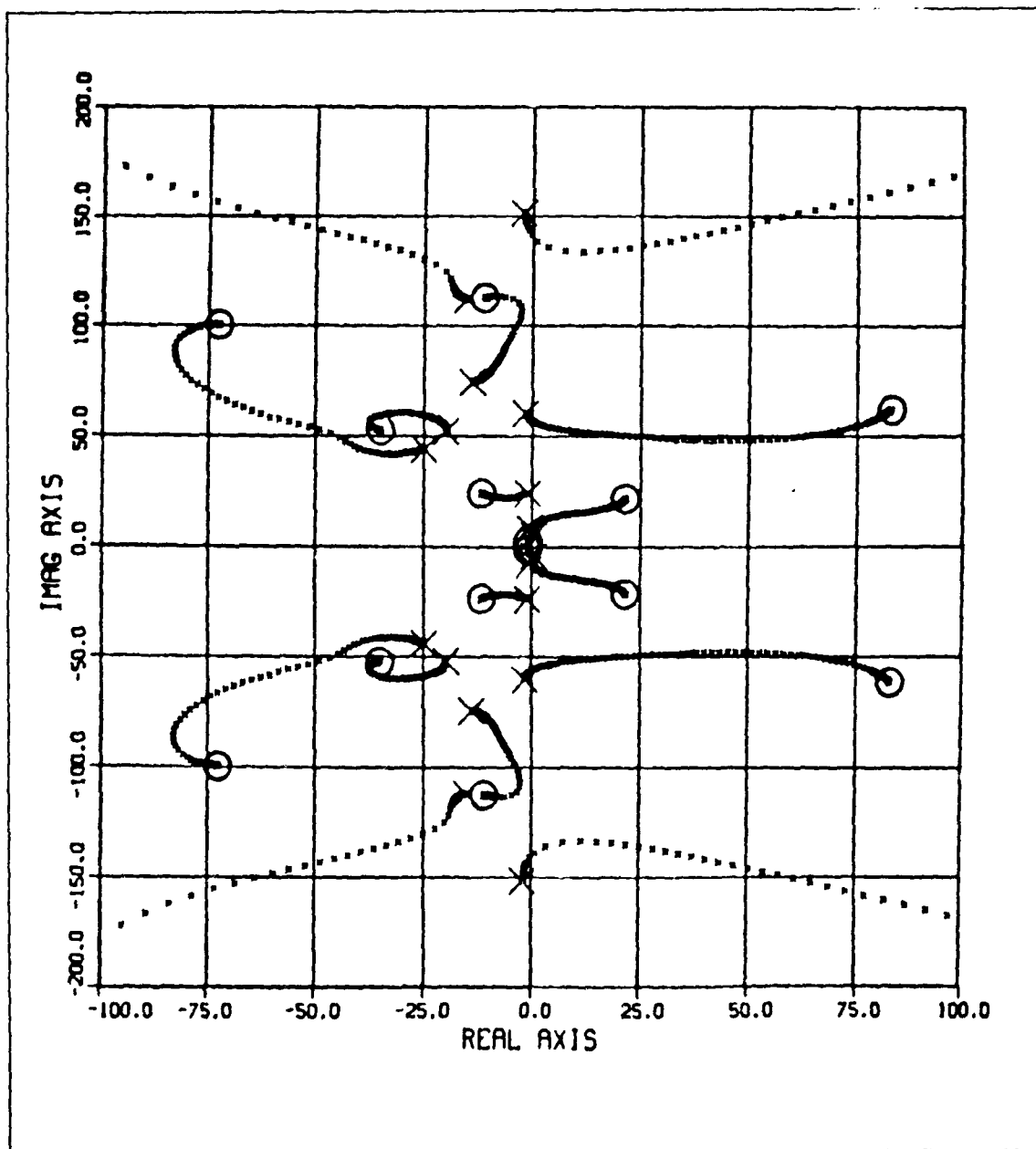


Figure 77. Root-Locus for First FM Result

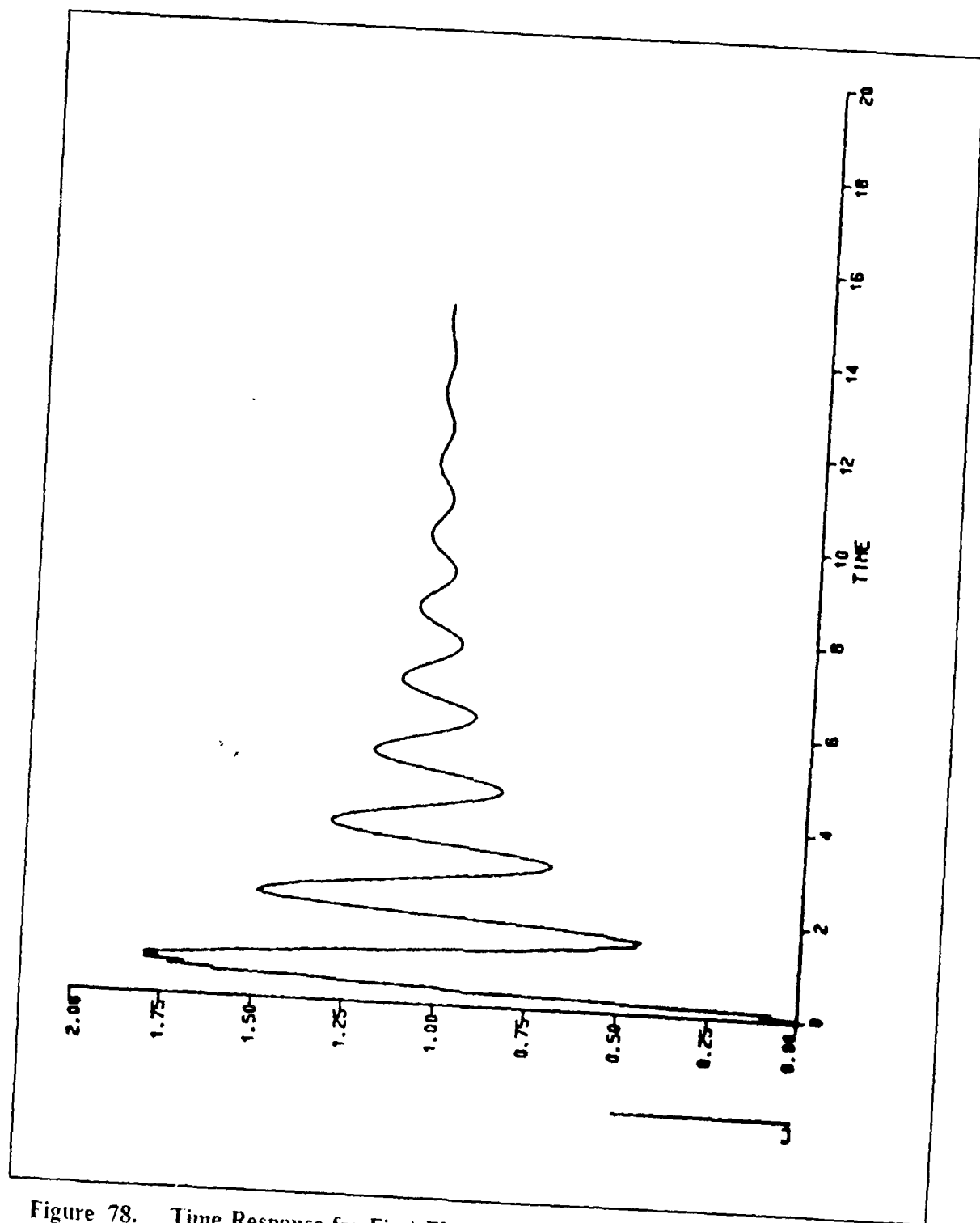


Figure 78. Time Response for First FM Result

Running the Function Minimization program but changing ζ and w_n to get an output with the desired settling time, we got the coefficients for this compensator as shown in Figure 79. $G_c(s)$ is the closed loop transfer function and

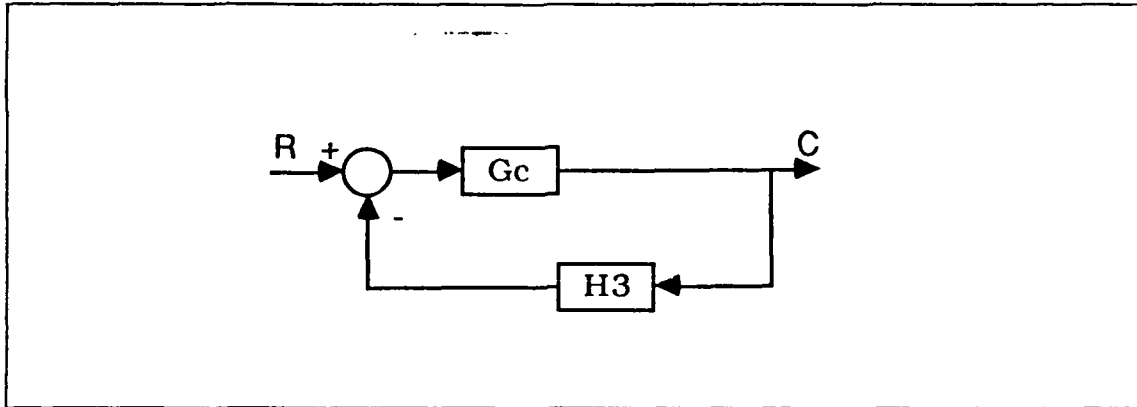


Figure 79. Second compensator Resulting from Function Minimization.

$$H_3(s) = \frac{8.11(s + 1.1531)(s + 4.878)}{(s + 3.672 \mp j5.67)} \quad (56)$$

The Root-Locus for this is given in Figure 80. Again by getting tabulated root locations we see that for a Root-Locus gain = 0.124 we get all roots in the left half plane. Then our final compensator becomes ;

$$H_3(s) = \frac{s^2 + 6.0311s + 5.625}{s^2 + 7.344s + 45.632} \quad (57)$$

The time response for this case is shown in Figure 81. As we see the settling time and amount of overshoot are within specific limits.

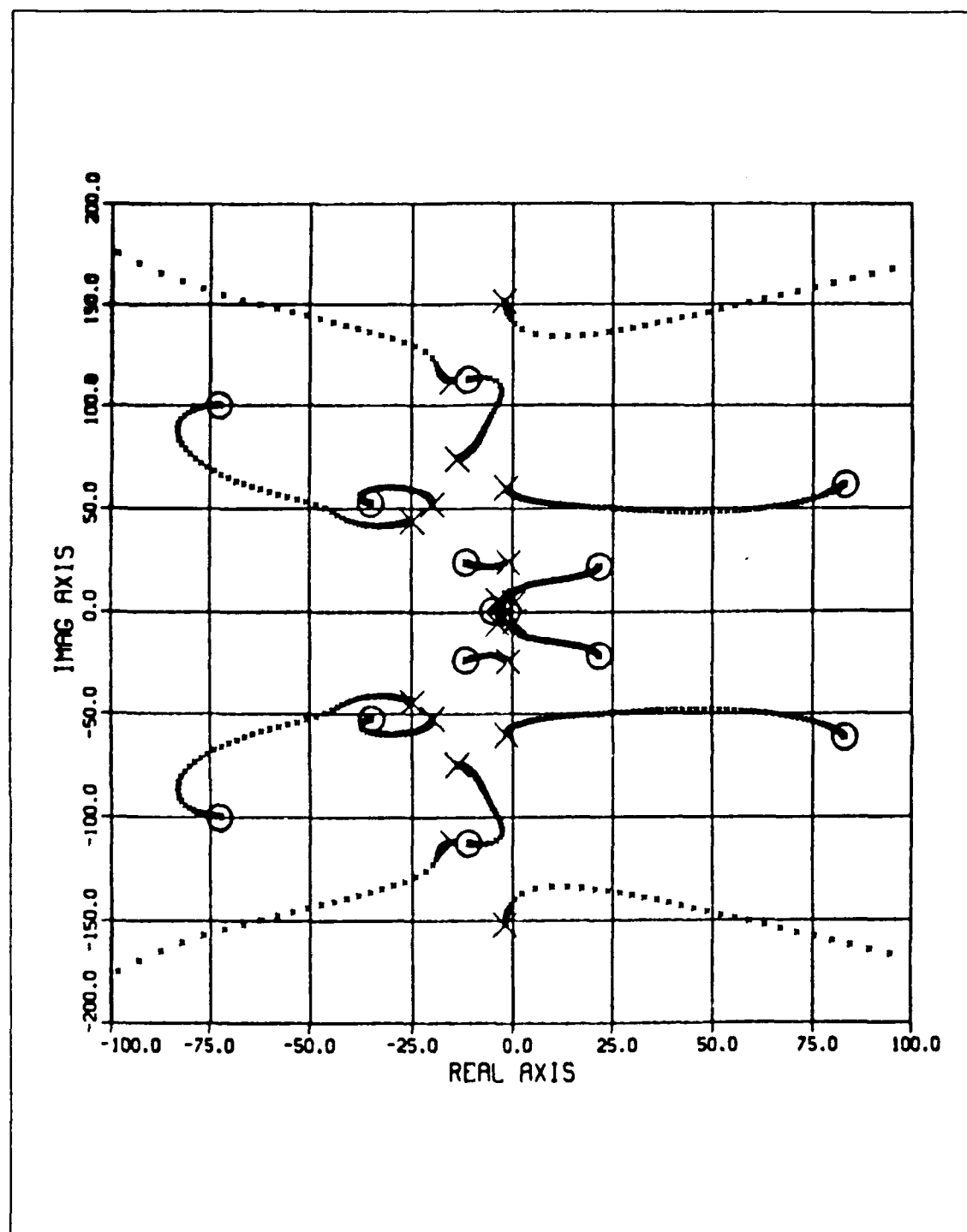


Figure 80. Root-Locus for Second FM Result

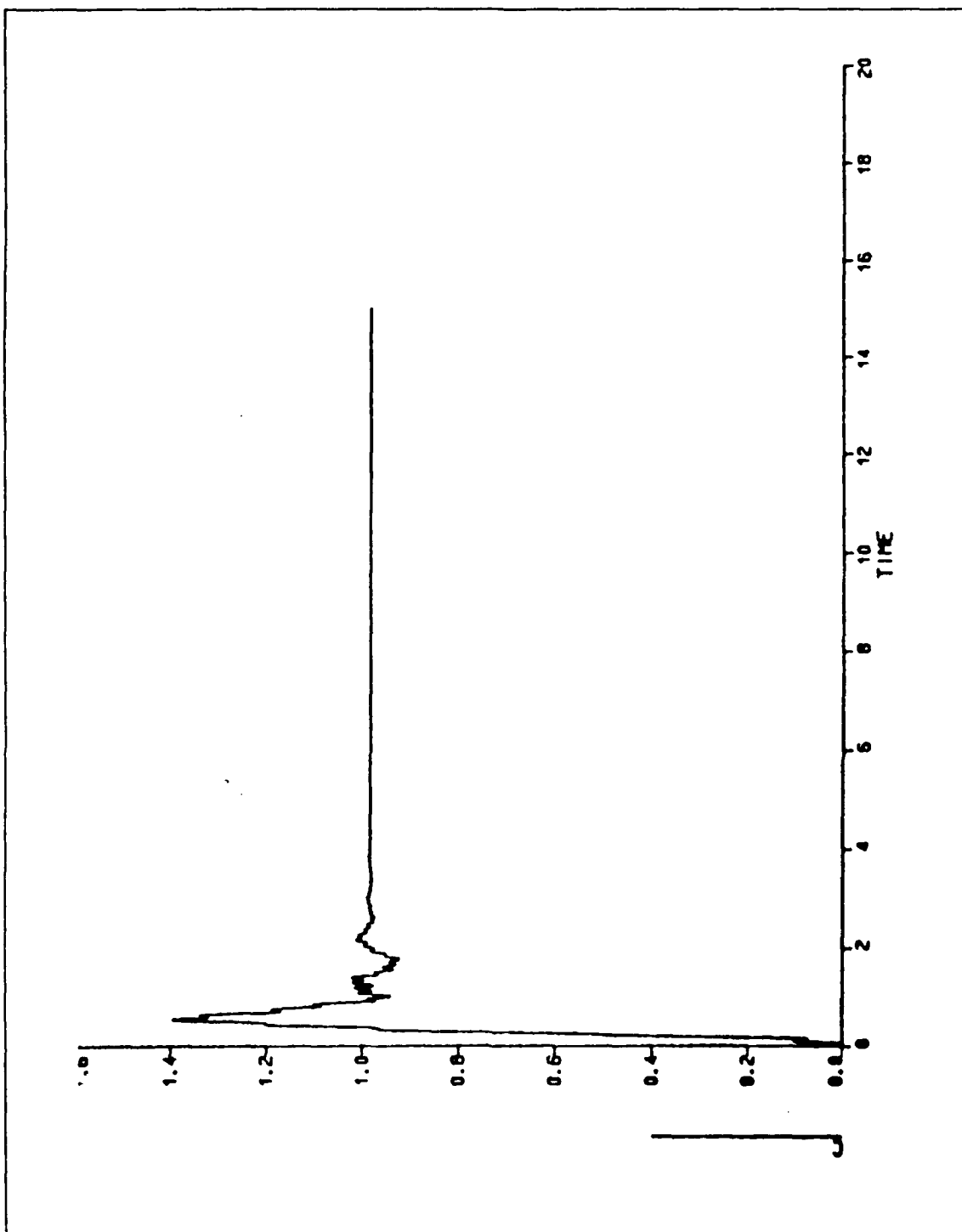


Figure 81. Time Response for Second FM Result

Then our total compensated system becomes as in Figure 82, where $G(s)$ is the Open loop transfer function and

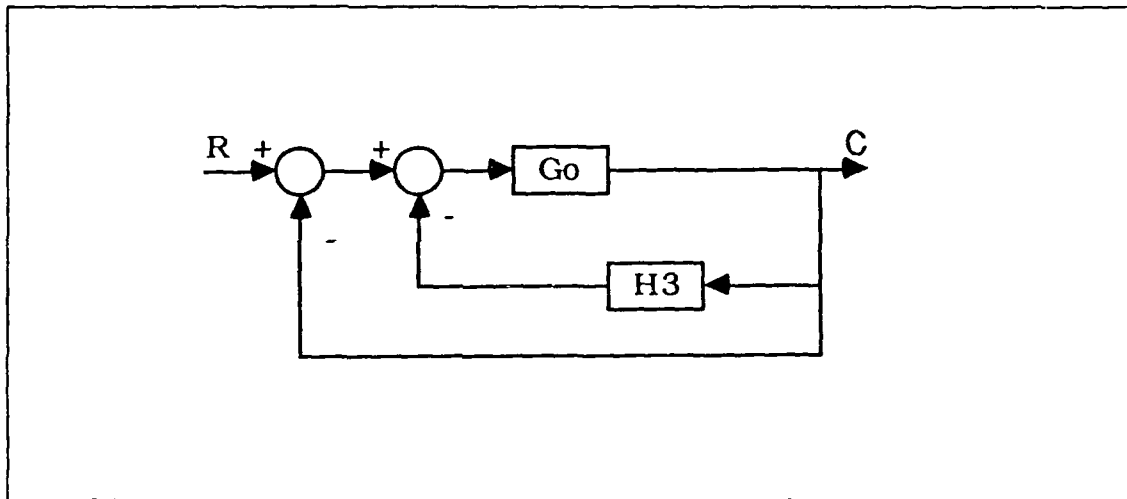


Figure 82. Block Diagram of Total Compensated System

$$H_3(s) = \frac{(s + 1.1531)(s + 4.878)}{(s + 3.672 \mp j5.67)} \quad (58)$$

As a result of this problem we may generalize that, if we have components close to the imaginary axis in both left and right half plane, we have to use pole-zero couples close to each other as a compensator. In the next chapter building filters by using pole placement and state feedback methods will be discussed.

IV. DESIGNING FEEDBACK COMPENSATORS USING POLE PLACEMENT

To compensate the systems by state feedback, we may use several computer programs to find the feedback coefficients. Sometimes observing or feeding back some of the states may not be possible. For such cases we might have to build an estimator to estimate the states.

The purpose of this chapter is to investigate whether it is possible to build a filter by using these feedback coefficients as zeros and adding extra poles to the feedback path.

To illustrate this we chose the following plants.

A. ALL COMPONENTS ARE IN THE LEFT HALF PLANE

For this case the open loop transfer function of the plant is ;

$$G(s) = \frac{1000}{s(s+1)(s+2)(s+10)} \quad (59)$$

We want to design a feedback compensator (a filter) to stabilize this system after determining the state feedback coefficients. Since the uncompensated system is fourth order, in order to have full state feedback we have to name four roots.

In this thesis we used the SVS (State Variable Systems) [Ref. 2] program.

To use the computer program we have to determine $n-1$ roots (where n is the system order) and find the fourth root from the characteristic equation to enter the program. Our desired roots are

$$s_{1,2} = -1 \mp j2, s_3 = -4 \quad (60)$$

Then $H(s)$ becomes

$$H(s) = s^3 + 6s^2 + 13s + 20 \quad (61)$$

If we get the $GH(s)$ to find the characteristic equation

$$GH(s) = \frac{1000(s^3 + 6s^2 + 13s + 20)}{s^4 + 13s^3 + 32s^2 + 20s} \quad (62)$$

The characteristic equation then becomes

$$s^4 + 1013s^3 + 6032s^2 + 13020s + 20000 \quad (63)$$

And the roots of the characteristic equation are ;

$$S_1 = -3.988, s_{2,3} = -0.997 \mp j1.996, s_4 = -1007.02 \quad (64)$$

The root locations obtained with this design are within 0.3 % of the desired values and the 4th root is at -1007.02 .

Now we may go to the SVS program to find the state feedback coefficients which would place the roots exactly at -4, -1 \mp j2, -1007.02. The state feedback coefficients we got from SVS are ;

$$H(s) = 0.9998s^3 + 5.982s^2 + 12.91s + 19.8 \quad (65)$$

These coefficients are negligibly different from the ones we obtained by multiplying the desired roots. For accuracy we used the coefficients that we got from SVS.

For all practical purposes we may accept the k_3 which is coefficient of s^3 , as 1. Our state feedback compensated system then becomes as in Figure 83, where $G(s)$ is the open loop transfer function and

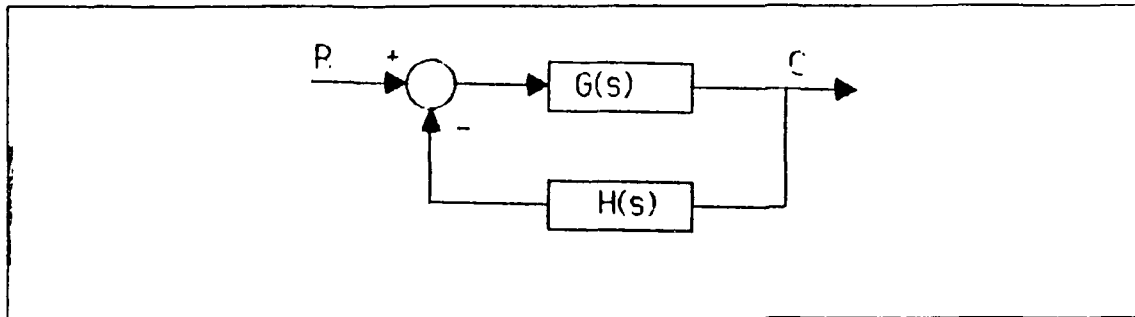


Figure 83. State Feedback Compensated System.

$$H(s) = s^3 + 5.982s^2 + 12.91s + 19.8 \quad (66)$$

$H(s)$ in factored form is

$$H(s) = (s + 0.996 \mp j1.993)(s + 3.997) \quad (67)$$

Root-Loci for the loop transfer function $GH(s)$ are given in Figure 84.

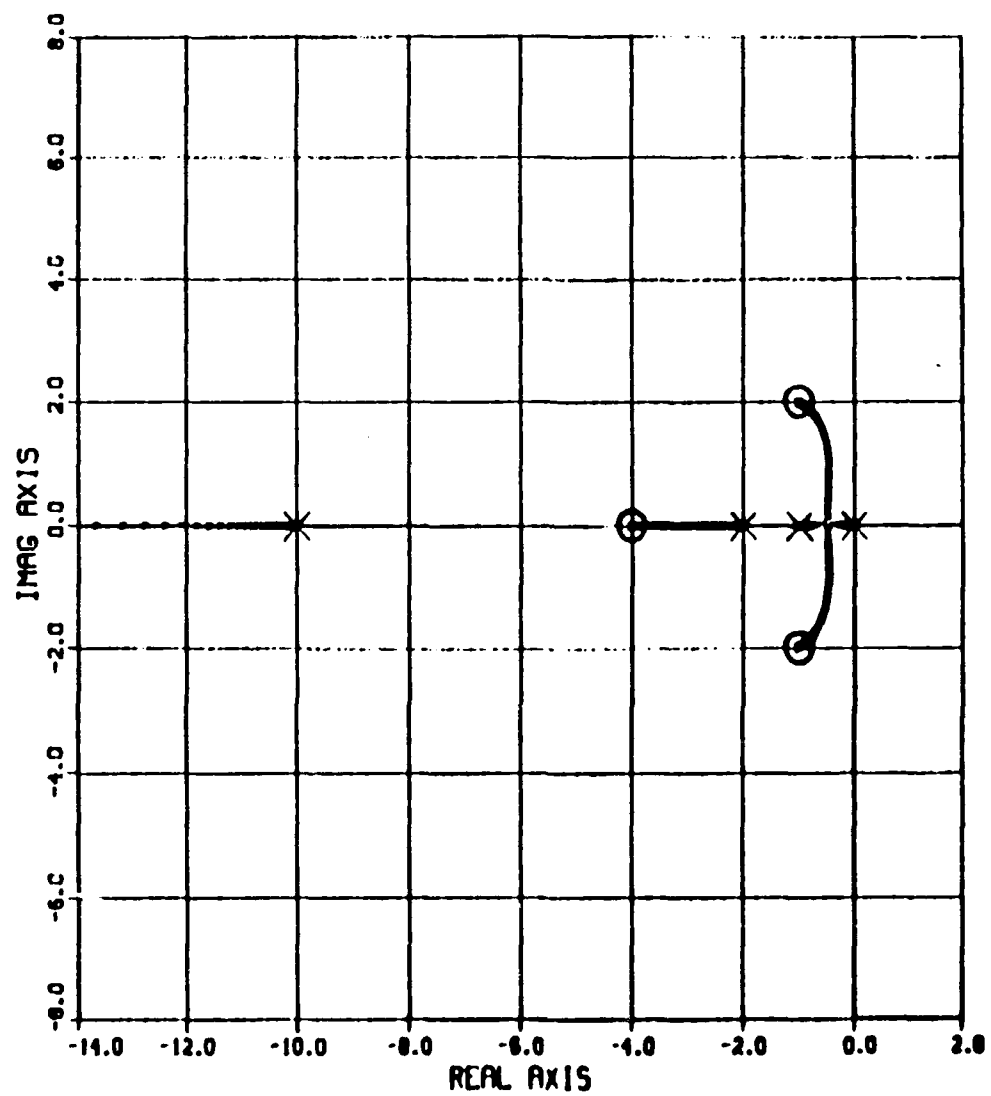


Figure 84. Root-Locus For State Feedback Loop Transfer Function.

To preserve the unity feedback we use block diagram manipulation and the system then becomes as in Figure 85.

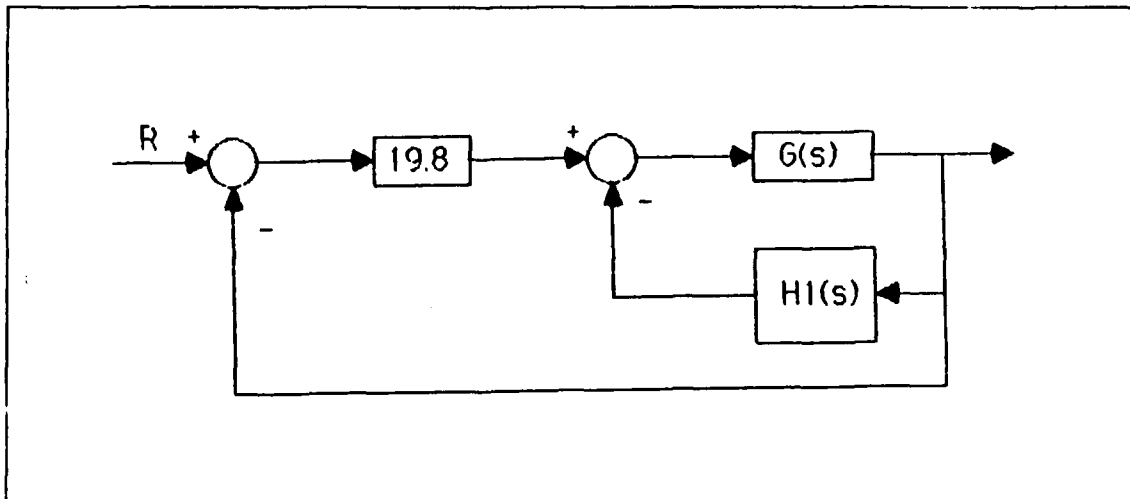


Figure 85. The Form of The System With Unity Feedback.

In Figure 85, $G(s)$ is the open loop transfer function and

$$H_1(s) = s(s + 2.99 \mp 1.993) \quad (68)$$

To convert the feedback compensator to make it more realizable as a filter, we have to choose the locations of the extra poles quite far away so they can not affect the transient behaviour of the system. If we put the extra poles into the system form shown in Figure 83 we get the required gain easily. The new system then becomes as in Figure 86, Where $G(s)$ is the open loop transfer function and

$$H_2(s) = \frac{K(s + 3.99)(s + 0.996 \mp j1.993)}{(s + 300)(s + 400)(s + 500)} \quad (69)$$

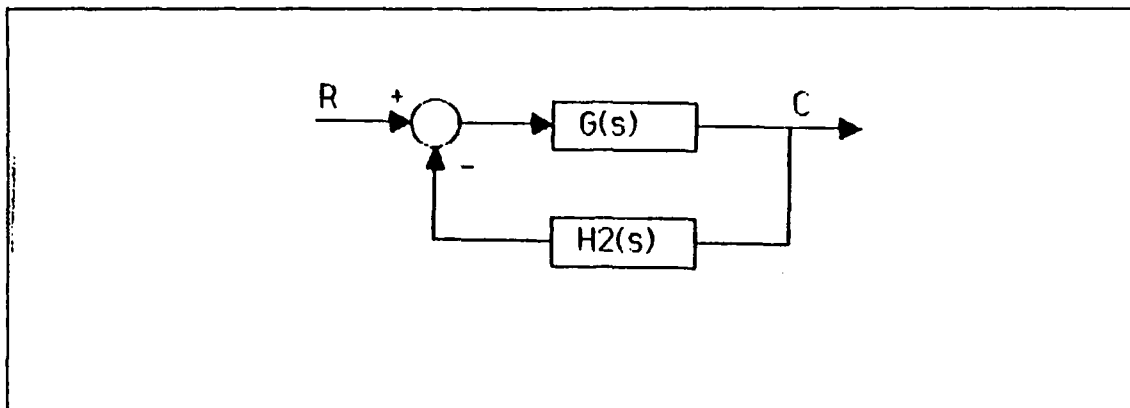


Figure 86. System With Extra Poles.

To find the value of the K we get the tabulated root locations of the Root-Locus for the loop transfer function $GH_2(s)$, which is shown in Figure 87.

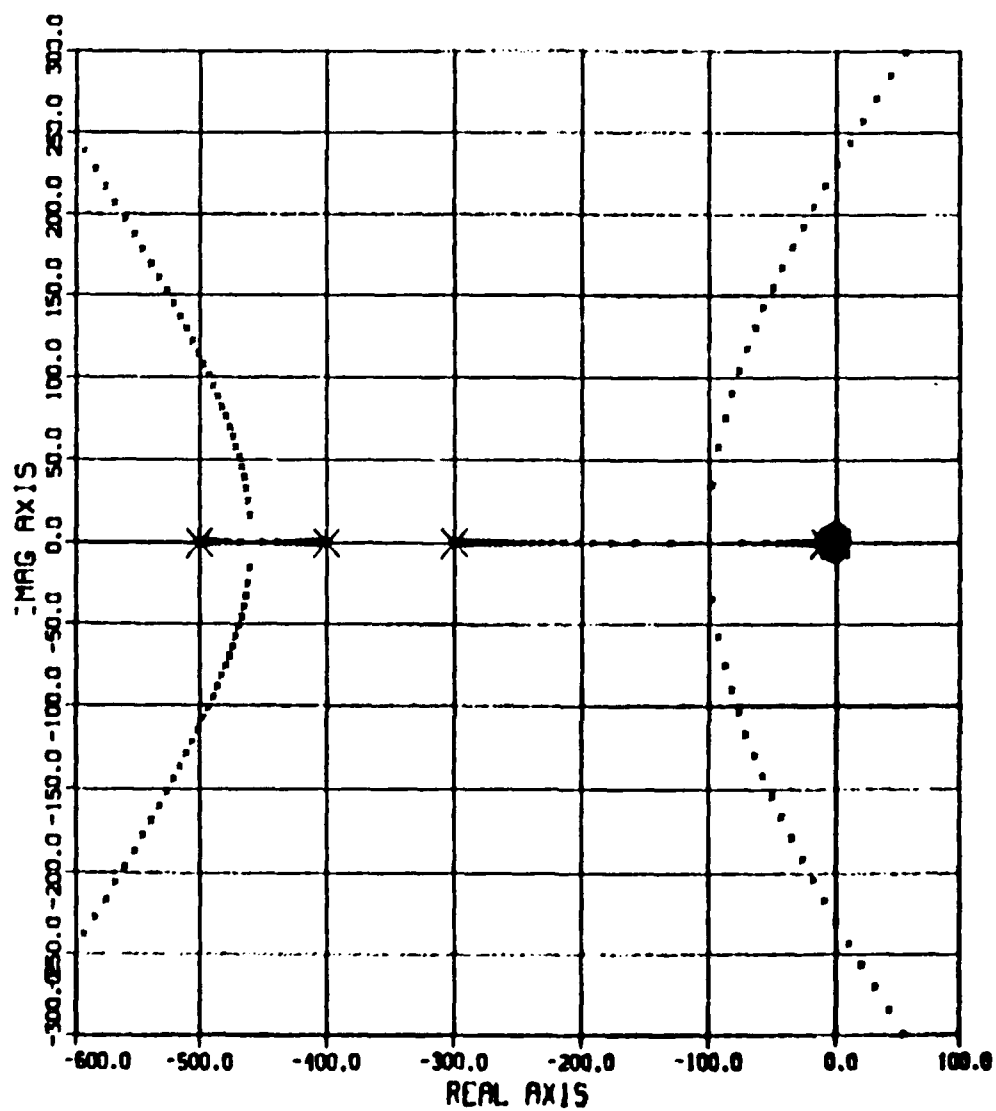


Figure 87. Root-Locus For Filter Feedback Loop Transfer Function.

Root-locus gain K to bring the roots where we want them was $K = 1.905459 \times 10^7$. By using the same manipulation as in Figure 85 if we redraw the block diagram of the system to preserve the unity feedback we get the final compensated system shown in Figure 88.

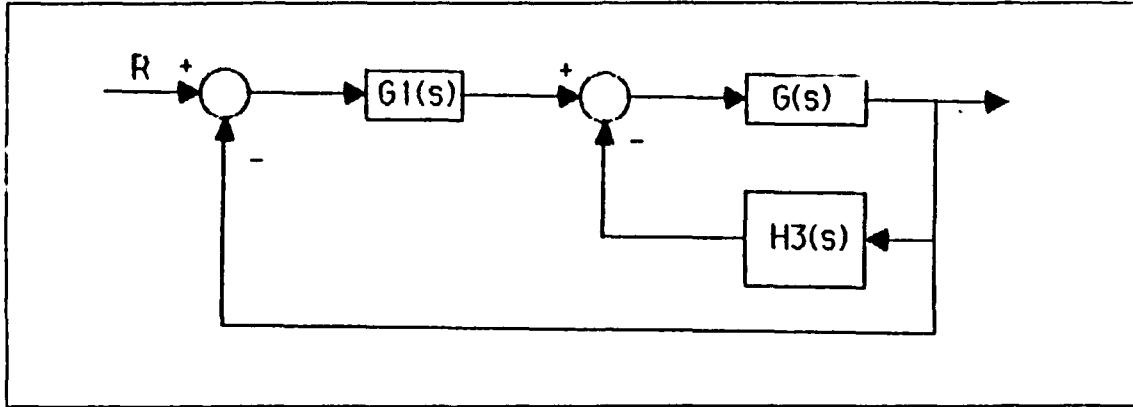


Figure 88. Final Compensated System

$G(s)$ is the open loop transfer function and

$$G_1(s) = \frac{19.8 \times 1.905459 \times 10^7}{(s + 300)(s + 400)(s + 500)} \quad (70)$$

$$H_3(s) = \frac{1.905459 \times 10^7 s(s + 2.99 \pm j1.99)}{(s + 300)(s + 400)(s + 500)} \quad (71)$$

The time responses for only state feedback and for the filter compensated case are Figures 89 and 90. As we may see from the transient responses since all roots are very close to the desired places the time responses are almost exactly same. The only disadvantage of this configuration may be to build two identical sets of poles to use in different places.

B. NON-MINIMUM PHASE SYSTEMS

As a second example we chose a non-minimum phase system to check if the idea works for both cases.

For this case the open loop transfer function of the plant is

$$G(s) = \frac{1000}{s(s - 1)(s + 2)(s + 20)} \quad (72)$$

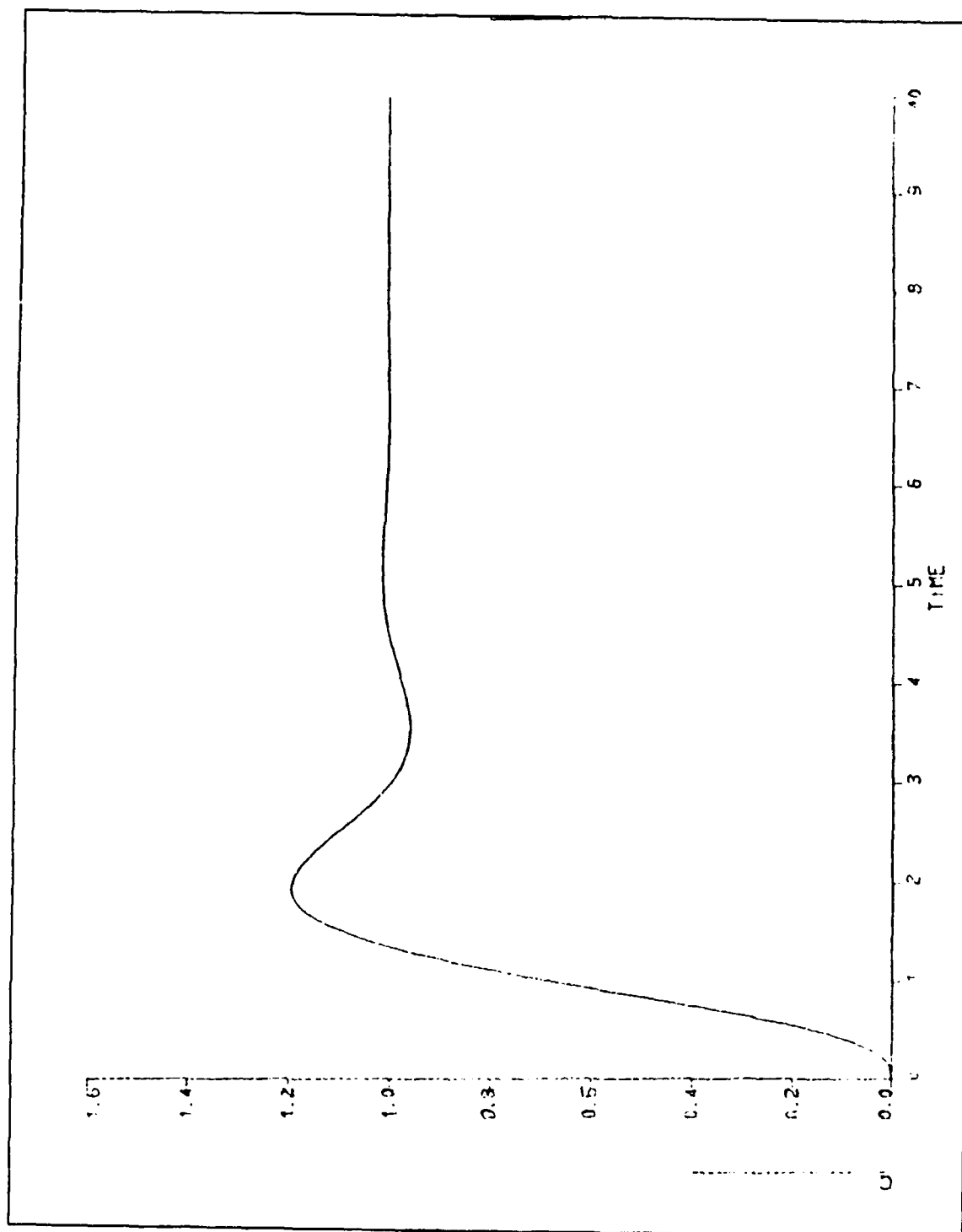


Figure 89. Time Response for Only State Feedback Compensated System

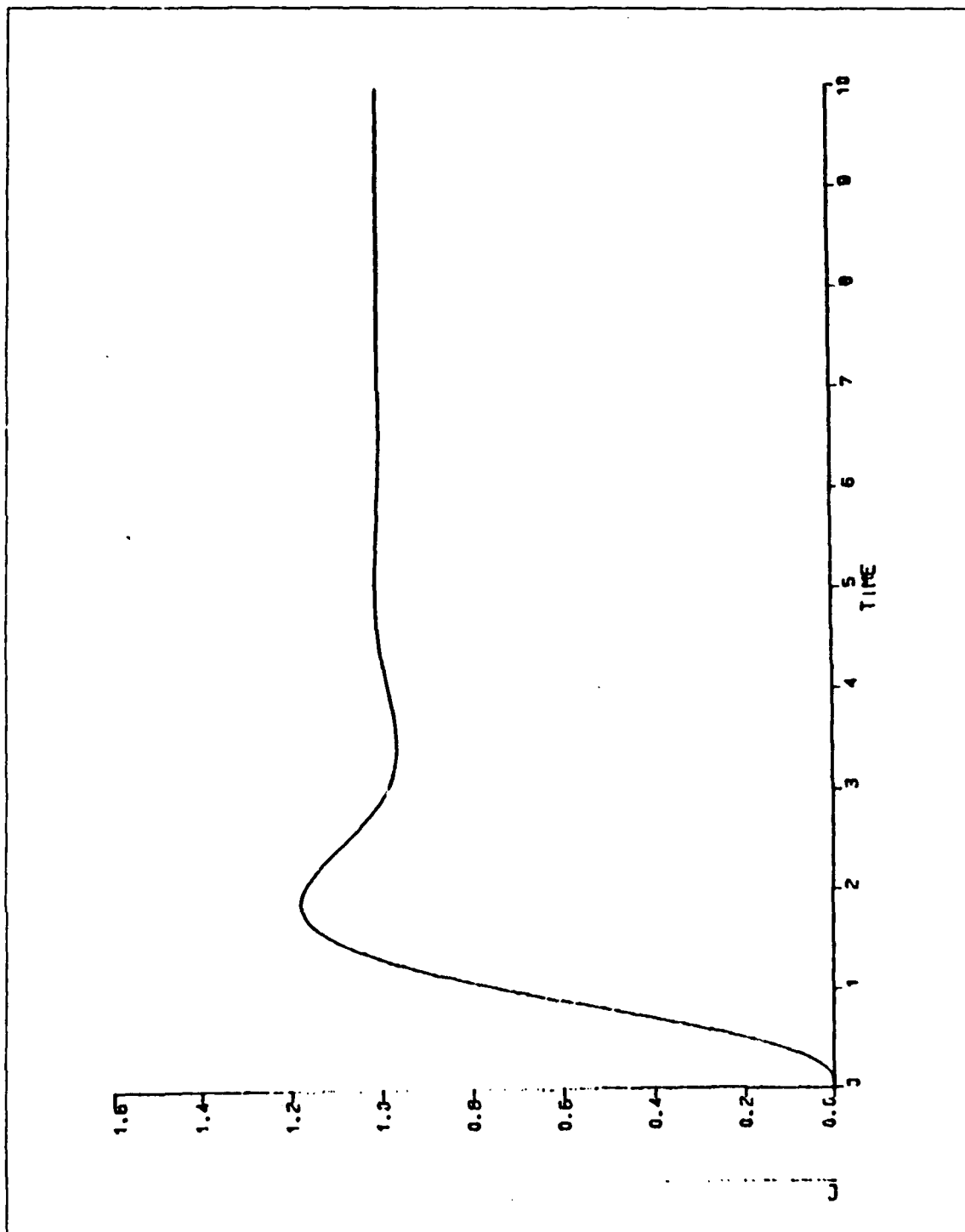


Figure 90. Time Response for Filter Compensated System

We want to design a feedback compensator (as a filter) to stabilize this system after determining the state feedback coefficients. Since the uncompensated system is fourth order, in order to have full state feedback we have to name four roots.

To use the computer program we have to determine $n-1$ roots (where n is the system order) and find the fourth root from the characteristic equation to enter the program. Our desired roots are ;

$$s_{1,2} = -1 \mp j1, s_3 = -3 \quad (73)$$

Then $H(s)$ becomes

$$H(s) = s^3 + 5s^2 + 8s + 6 \quad (74)$$

If we get the $GH(s)$ to find the characteristic equation

$$GH(s) = \frac{1000(s^3 + 5s^2 + 8s + 6)}{s^4 + 21s^3 + 18s^2 - 40s} \quad (75)$$

The characteristic equation then becomes

$$s^4 + 1021s^3 + 5018s^2 + 7960s + 6000 \quad (76)$$

And the roots of the characteristic equation are

$$s_1 = -2.967, s_{2,3} = -0.988 \mp j1.01, s_4 = -1016 \quad (77)$$

The root locations obtained with this design are within 1.1 % of the desired values and the 4th root is at -1016.

Now we may go to the SVS program to find the state feedback coefficients which would place the roots exactly at $-1 \mp j1, -3, -1016$.

The state feedback coefficients we got from SVS are

$$H(s) = 0.9999s^3 + 4.995s^2 + 7.952s + 5.956 \quad (78)$$

These coefficients again are negligibly different from the $H(s)$ we first accepted but we used the ones we found from SVS.

For all practical purposes we may accept the k_3 which is coefficient of s^3 as 1. Our state feedback compensated system then becomes as in Figure 91.

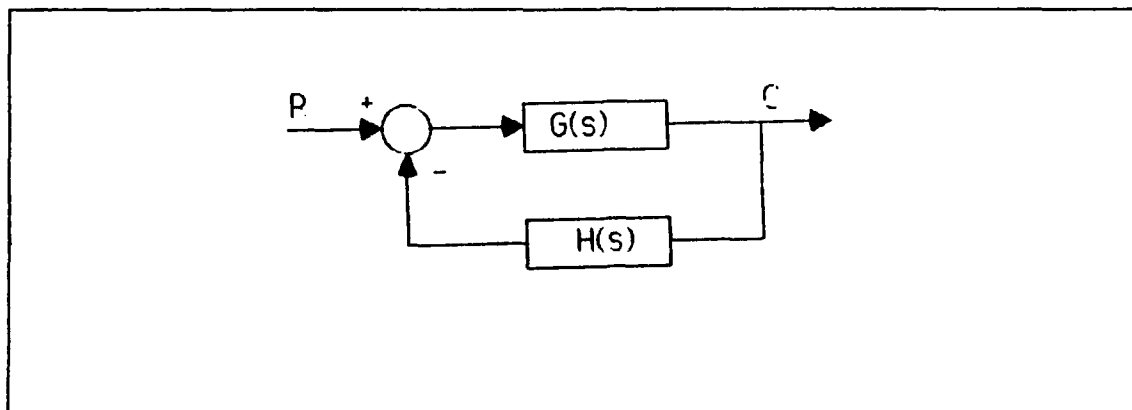


Figure 91. State Feedback to Compensate The System

$G(s)$ is the open loop transfer function and

$$H(s) = s^3 + 4.995s^2 + 7.952s + 5.956 \quad (79)$$

$H(s)$ in factored form is

$$H(s) = (s + 0.995 \mp j0.998)(s + 2.998) \quad (80)$$

The Root-Locus for the loop transfer function $GH(s)$ is given in Figure 92.

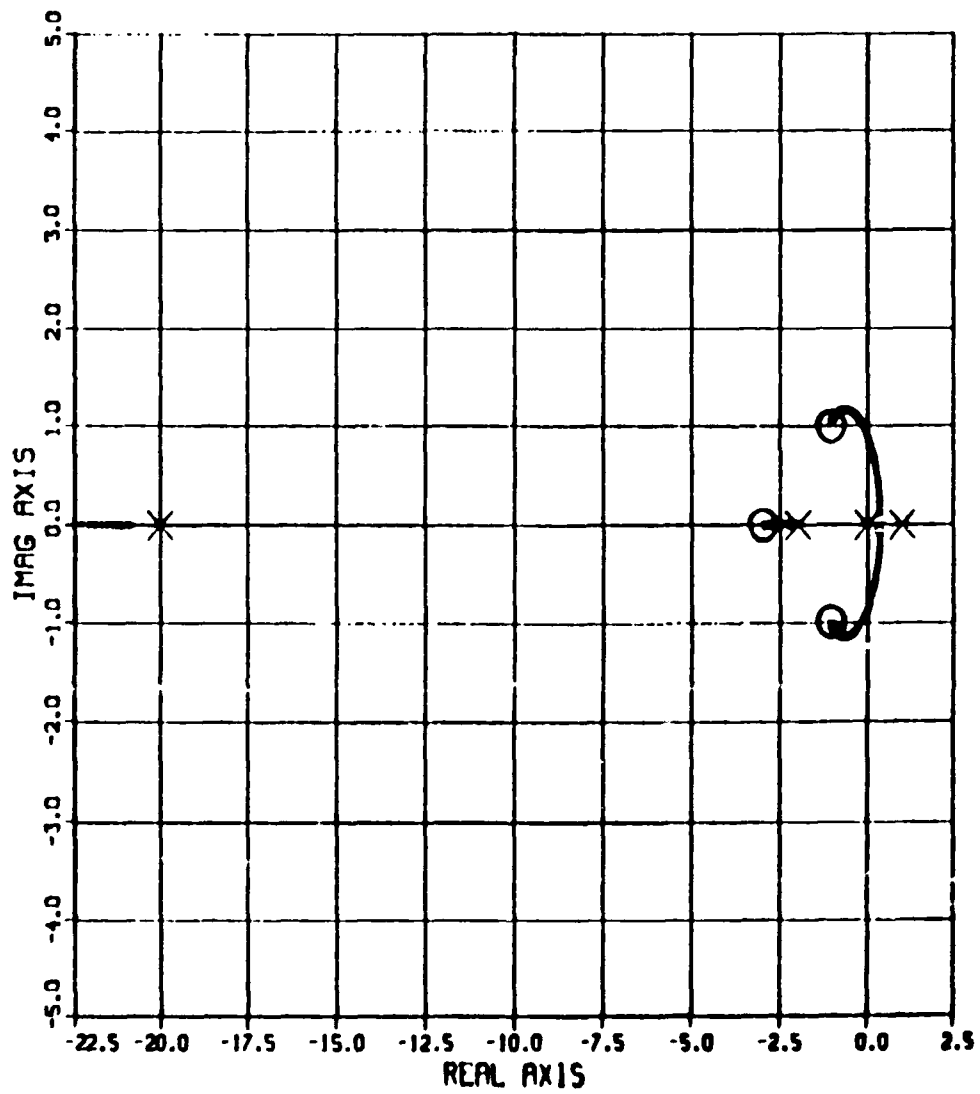


Figure 92. Root-Locus For State Feedback Loop Transfer Function.

To preserve unity feedback we may use block diagram manipulation. The next form of the system then becomes as in Figure 93 where $G(s)$ is the open loop transfer function and

$$H_1(s) = s(s + 2.494 \mp 1.316) \quad (81)$$

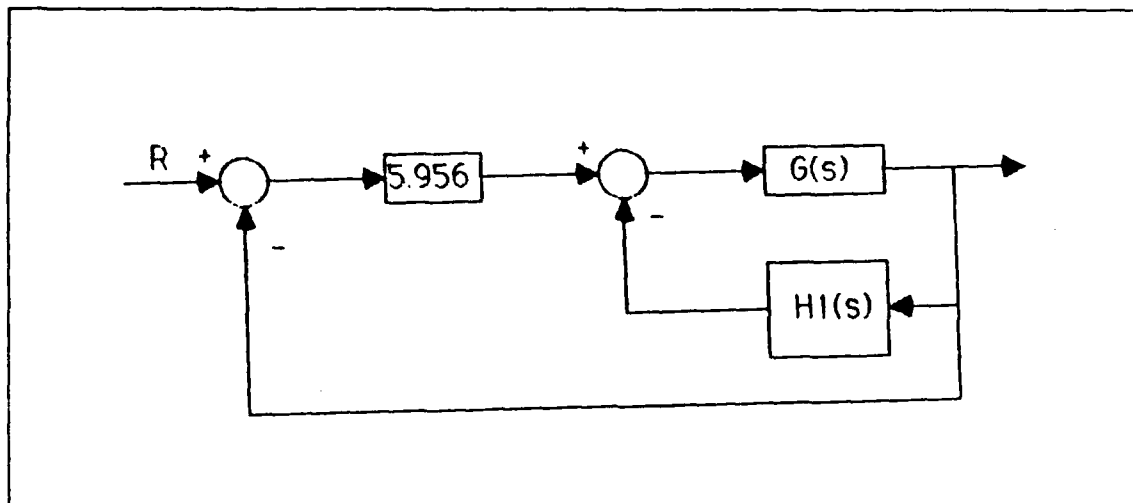


Figure 93. The Form of The System With Unity Feedback.

To convert the feedback compensator to make it more realizable as a filter, we have to choose the locations of the extra poles quite far away so they can not affect the transient behaviour of the system. If we put the extra poles into the system form shown in Figure 91 we get the required gain easily. The new system then becomes as in Figure 94.

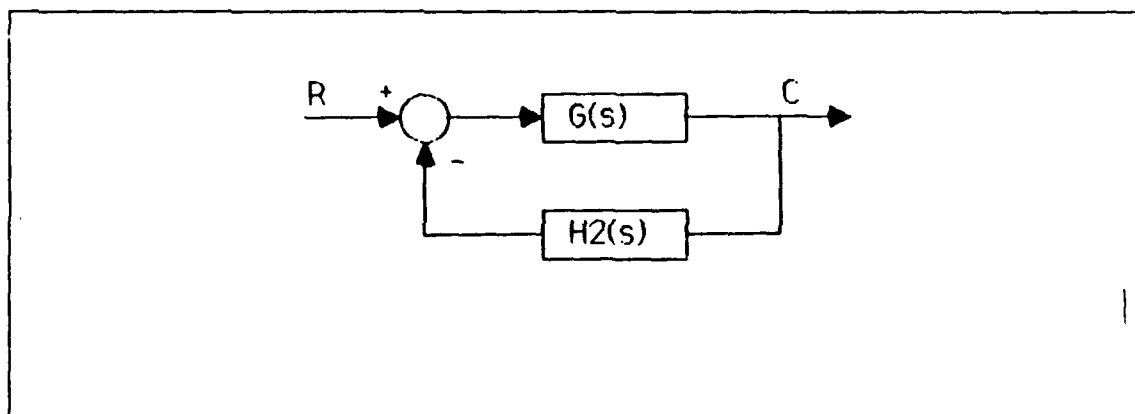


Figure 94. System With Extra Poles

$G(s)$ is the open loop transfer function and

$$H_2(s) = \frac{K(s + 2.998)(s + 0.995 \mp j0.998)}{(s + 300)(s + 400)(s + 500)} \quad (82)$$

To find the value of the K we get the tabulated root locations of the Root-Locus for the loop transfer function $GH_2(s)$, which is shown in Figure 95.

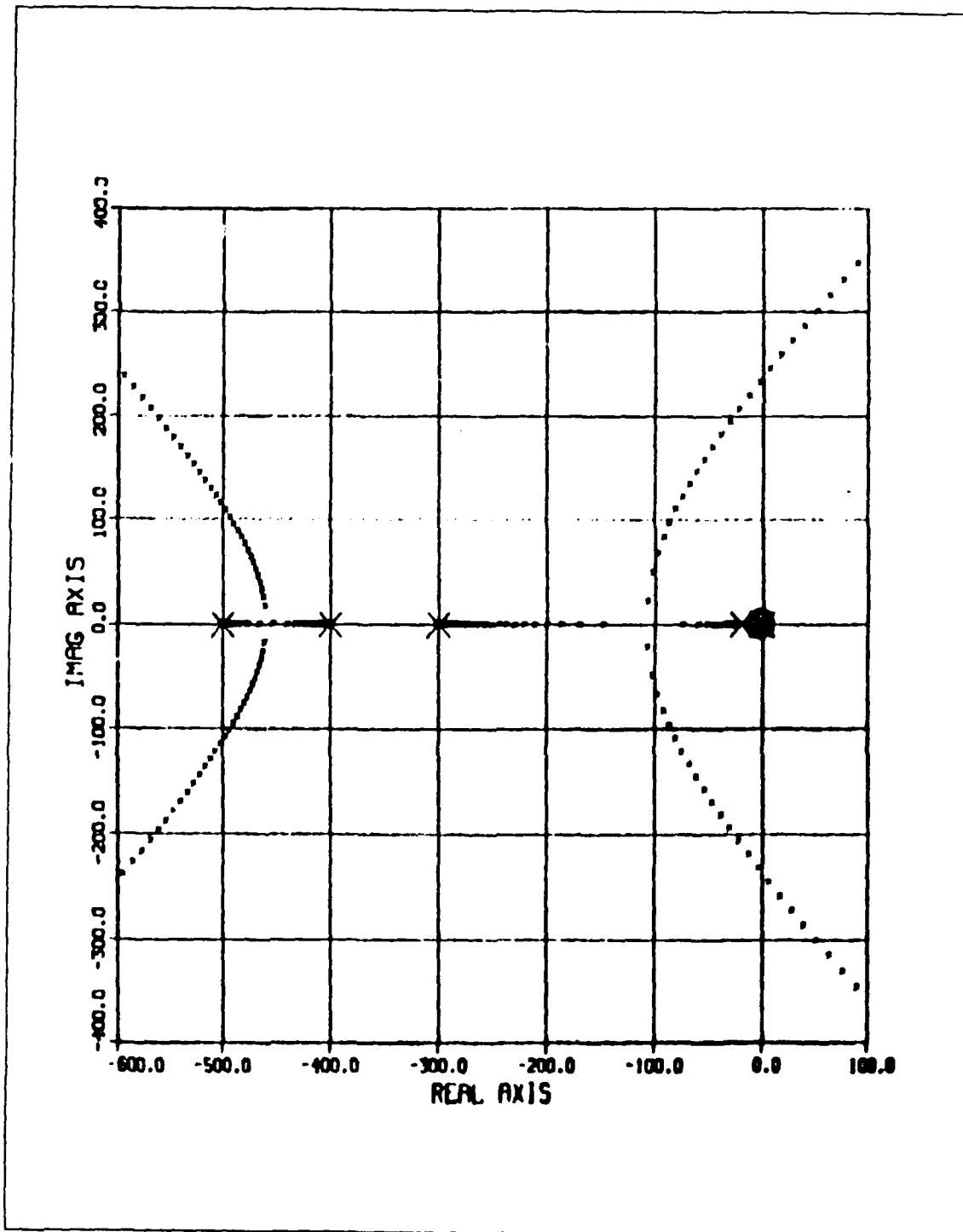


Figure 95. Root-Locus For Filter Feedback Loop Transfer Function.

The Root-locus gain K to bring the roots where we want them was $K = 1.1257119 \times 10^7$. By using the same manipulation as in Figure 93 if we redraw the block diagram of the system to preserve the unity feedback we get the final compensated system shown in Figure 96.

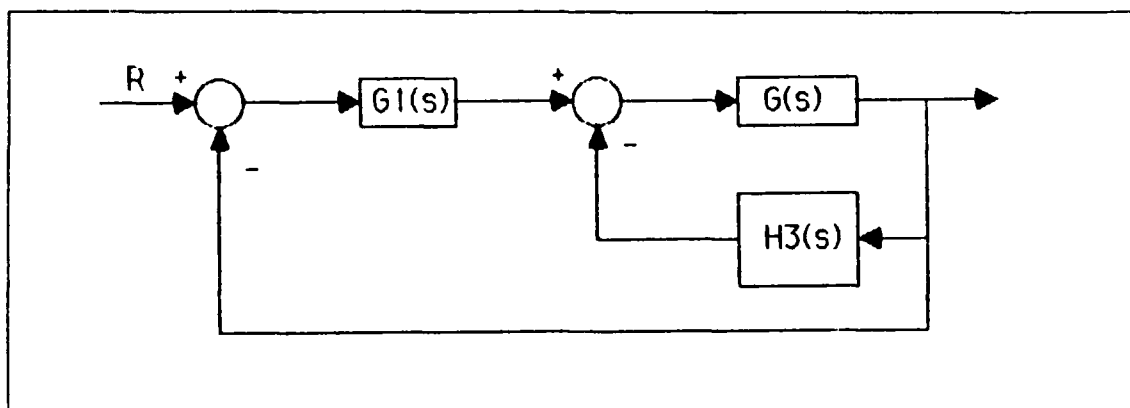


Figure 96. Final Compensated System

$G(s)$ is the open loop transfer function and

$$G_1(s) = \frac{5.956 \times 1.1257119 \times 10^7}{(s + 300)(s + 400)(s + 500)} \quad (83)$$

$$H_3(s) = \frac{1.1257119 \times 10^7 s(s + 2.494 \pm j1.316)}{(s + 300)(s + 400)(s + 500)} \quad (84)$$

The time responses for state feedback and for the filter compensated case are given Figures 97 and 98. As we may see from the transient responses since all roots are very close to the desired places the time responses are almost exactly same. The only disadvantage of this configuration may be to build two identical sets of poles to use in different places.

C. EXTENDING THE IDEA TO CASCADE COMPENSATION

The procedure for root placement with the feedback filter used only the loop transfer function. We may compensate the system with the same components and same gain by using cascade compensation. If we investigate the systems, the characteristic equations are the same with the filter in either the feedback path or in the forward path. Except this time instead of the poles, the zeros of the filter are the zeros of the closed loop

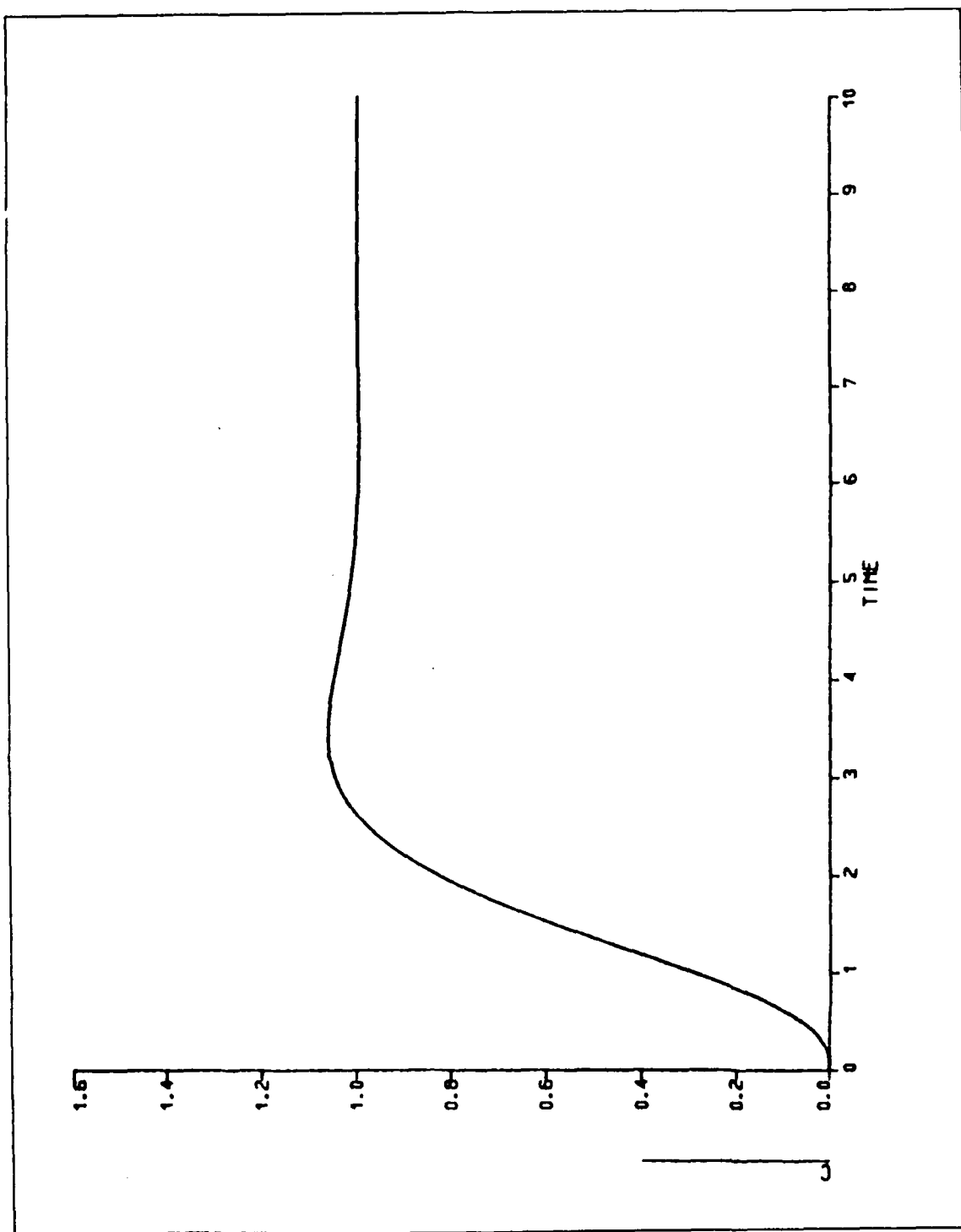


Figure 97. Time Response For Only State Feedback Compensated System

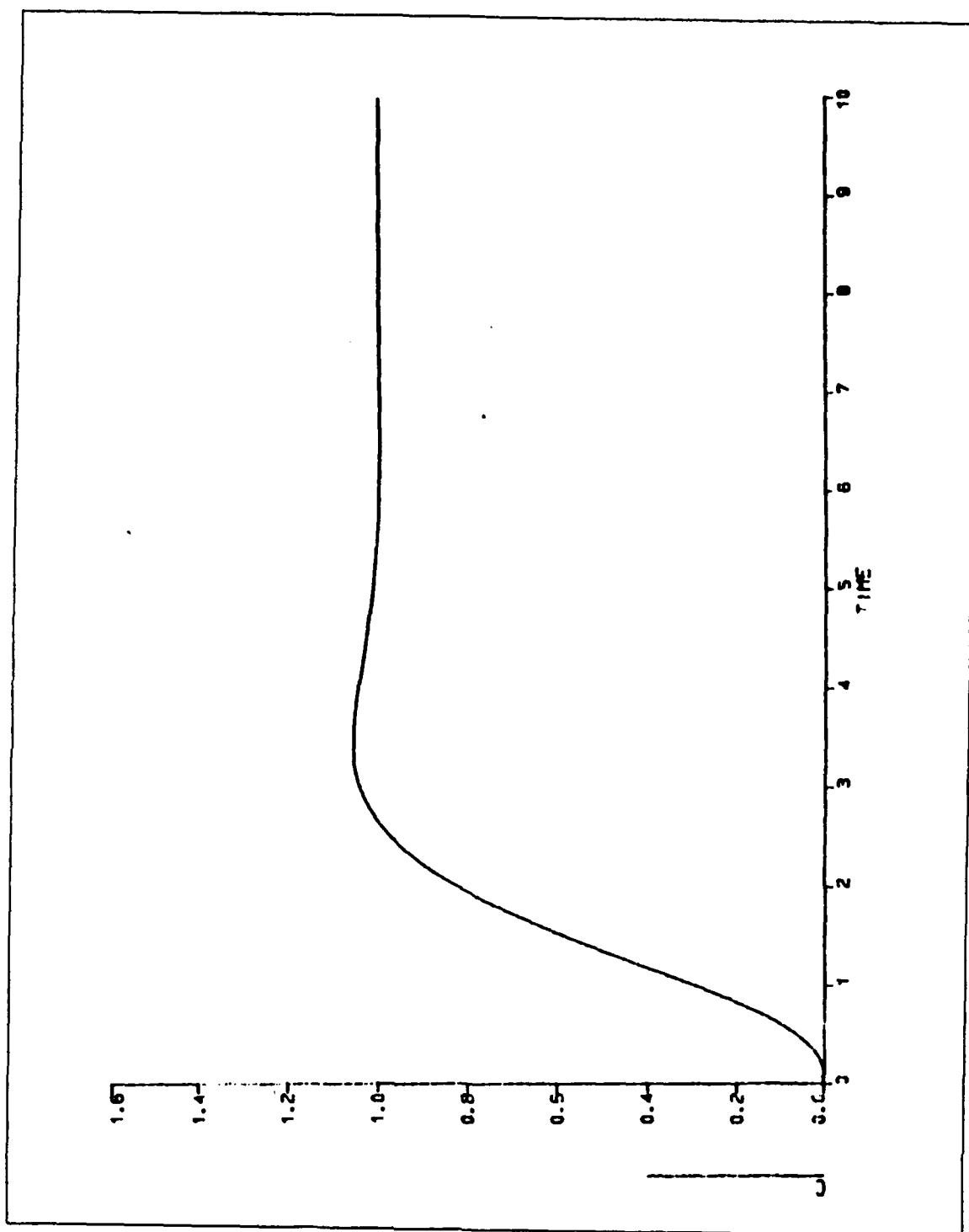


Figure 98. Time Response For Filter Compensated System

transfer function but the roots of the closed loop transfer function are in the same places as with the feedback compensation scheme.

To show the result of this idea we chose the same plants. For the first one if we put the same filter in the forward path the system becomes as in Figure 99, where

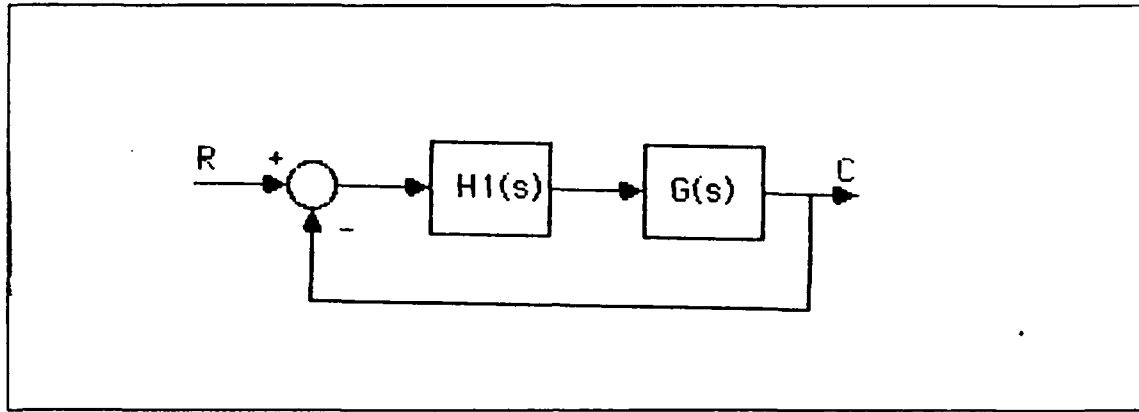


Figure 99. Cascade Compensated System

$$G(s) = \frac{1000}{s(s+1)(s+2)(s+10)} \quad (85)$$

$$H_1(s) = \frac{1.905459 \times 10^7 (s+1 \mp j2)(s+4)}{(s+300)(s+400)(s+500)} \quad (86)$$

The time response for this system is given in Figure 100. As we may see the time responses and oscillatory frequencies are almost identical with the feedback compensation time response which is shown in Figure 90. In this configuration because of the derivatives due to the cascade zeros we have faster rise time.

We checked the roots of the systems for both the feedback and the cascade configuration. They were identical within 0.37 % difference.

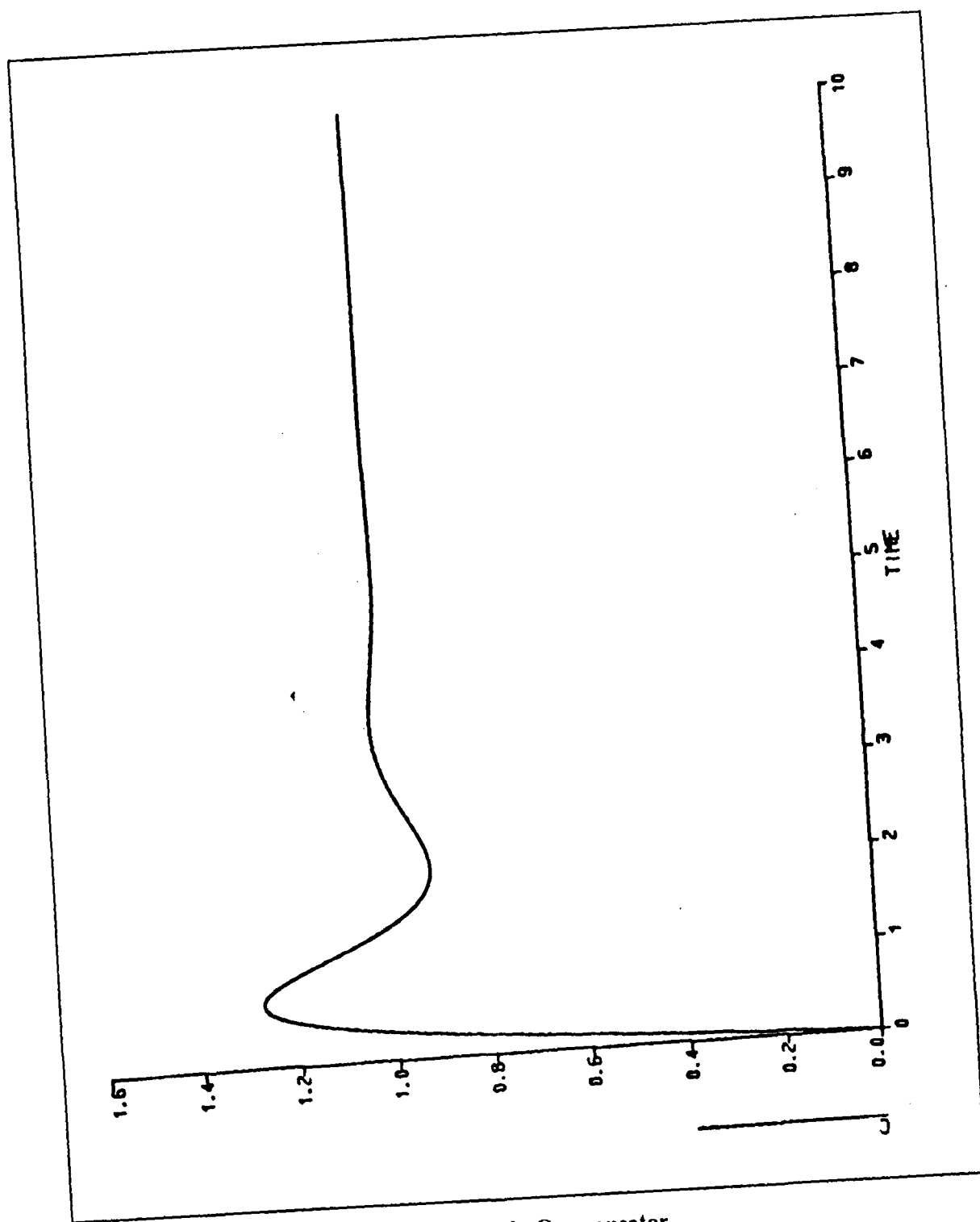


Figure 100. Time Response for Cascade Compensator

The second plant becomes as in Figure 101 if we put the compensator in the forward path, where

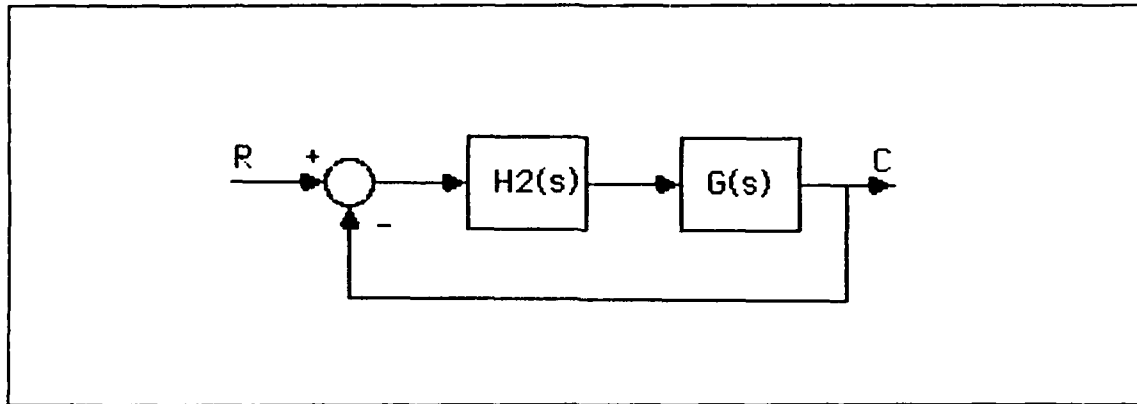


Figure 101. Cascade Compensated Second System

$$G(s) = \frac{1000}{s(s-1)(s+2)(s+20)} \quad (87)$$

$$H_2(s) = \frac{1.1257119 \times 10^7 (s+1 \mp j)(s+3)}{(s+300)(s+400)(s+500)} \quad (88)$$

The time response for this case is given in Figure 102. Again as we may see the time response is essentially the same as that of the feedback configuration which is shown in Figure 98. Here again we have a faster rise time because of derivatives in the cascade filter.

As we may see from the Figures 99 and 100 we put the filter before the plant in both cases and we used the DSL (Dynamic Simulation Language) which behaves the same way with the hardware.

Then we decided to simulate the systems with ALCON (Simulation program for personal computers). This program uses arithmetic polynomial calculations to calculate the step responses for the systems.

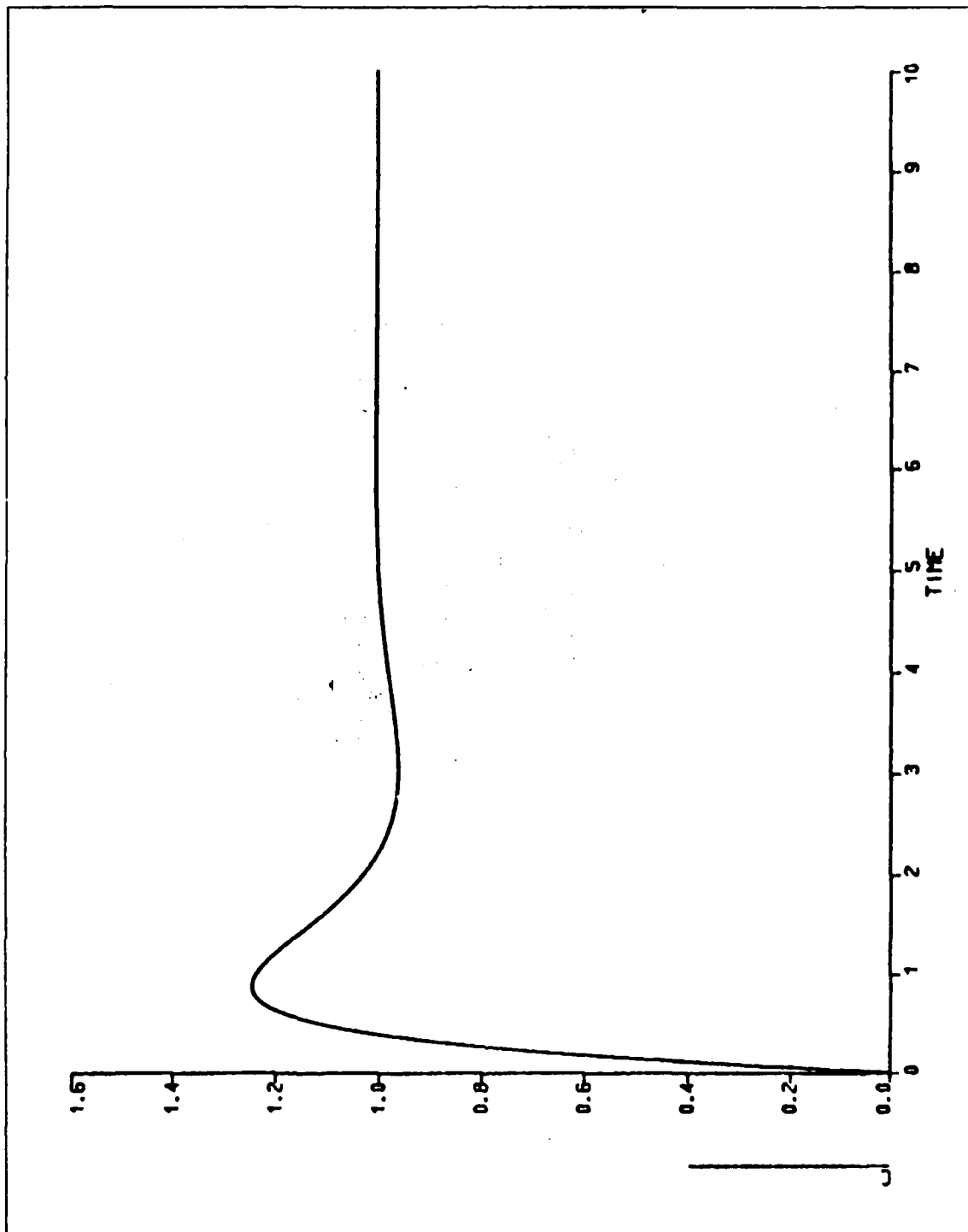


Figure 102. Time Response For Cascade Compensator (Second example)

The resulting time responses from this program for both problems are given in Figures 103 and 104. As we see high frequency ripples exist for a short time period which we didn't see in the DSL simulation.

To investigate the reason for this difference we interchanged the blocks in Figures 99 and 100, that is, we fed the error signal into the G block and the output from the G block into the H block. For this combination resulting step responses are given in Figures 105 and 106 which are identical with the time responses we obtained from ALCON.

For these DSL simulation cases, in the first case the filter is before plant, the plant works as Low-Pass filter and the high frequency ripples are not present, but in the second case the filter follows plant, and the ripples appear in the output.

We could not decide which one of the simulations is true. According to linear theory, changing the places of the plants should not affect to the behaviour of the system.

To investigate the reason for this difference is left as a recommendation.

As a result of these examples we may generalize the design steps as follows :

1. Select $N - 1$ desired roots.
2. Multiply them and find the $H(s)$.
3. Find $GH(s)$ and get the characteristic equation.
4. Find the roots of the characteristic equation. If they are acceptably close to the desired roots, use $H(s)$ as a state feedback directly, otherwise go to SVS or another computer program to find the state feedback coefficients to put the roots exactly at the desired places.
5. Choose the extra poles as far away as possible and put them in the denominator of $H(s)$.
6. Draw the Root-Locus for the loop transfer function $GH(s)$ (including the extra poles) and get the tabulated data.
7. Select the appropriate gain to have the system roots in the desired places (generally this will be very high gain).
8. If the purpose is to design a cascade filter, put the resulting filter directly in the forward path. Otherwise perform the block diagram manipulation as shown in the examples and preserve the unity feedback.

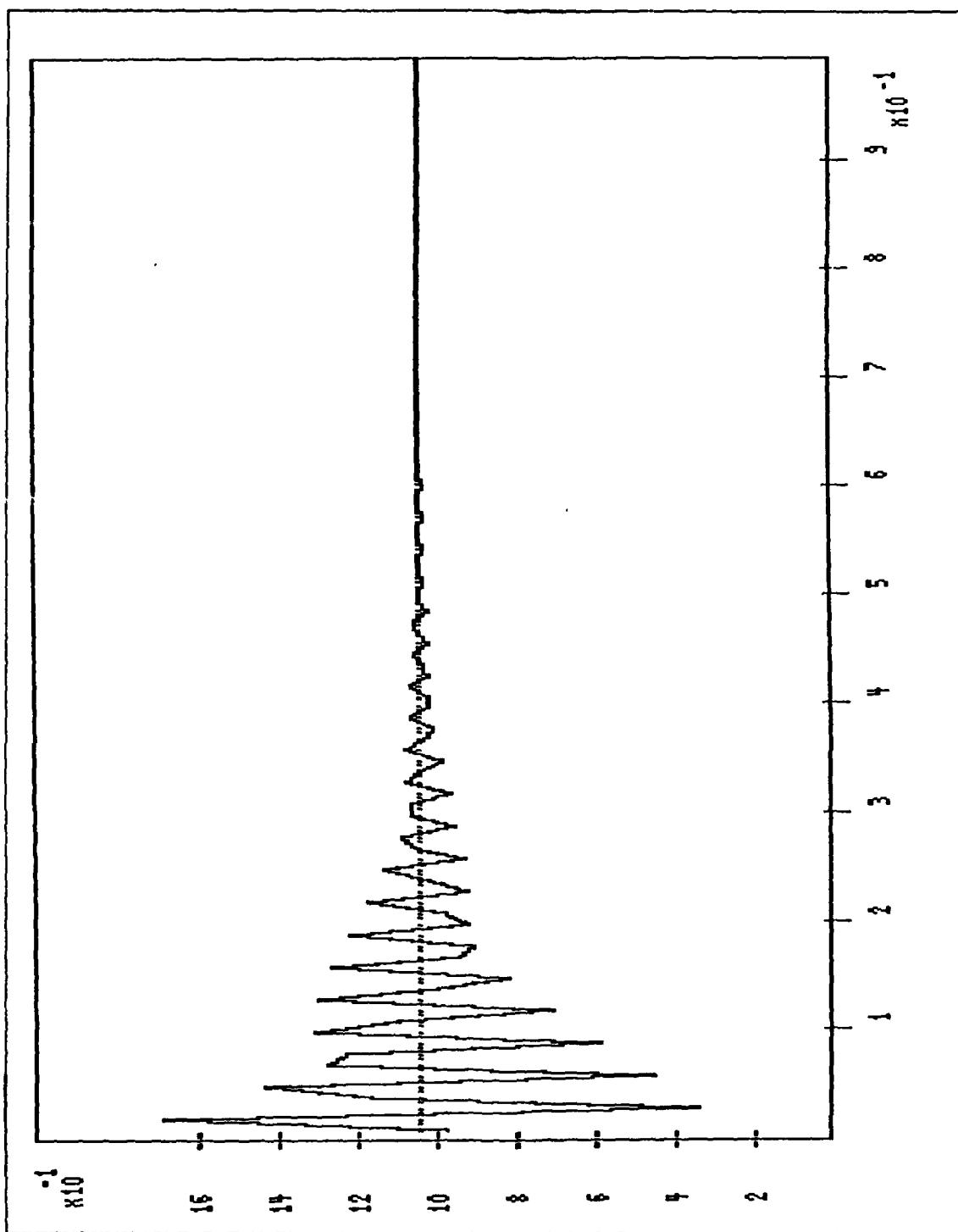


Figure 103. ALCON Simulation of First Problem

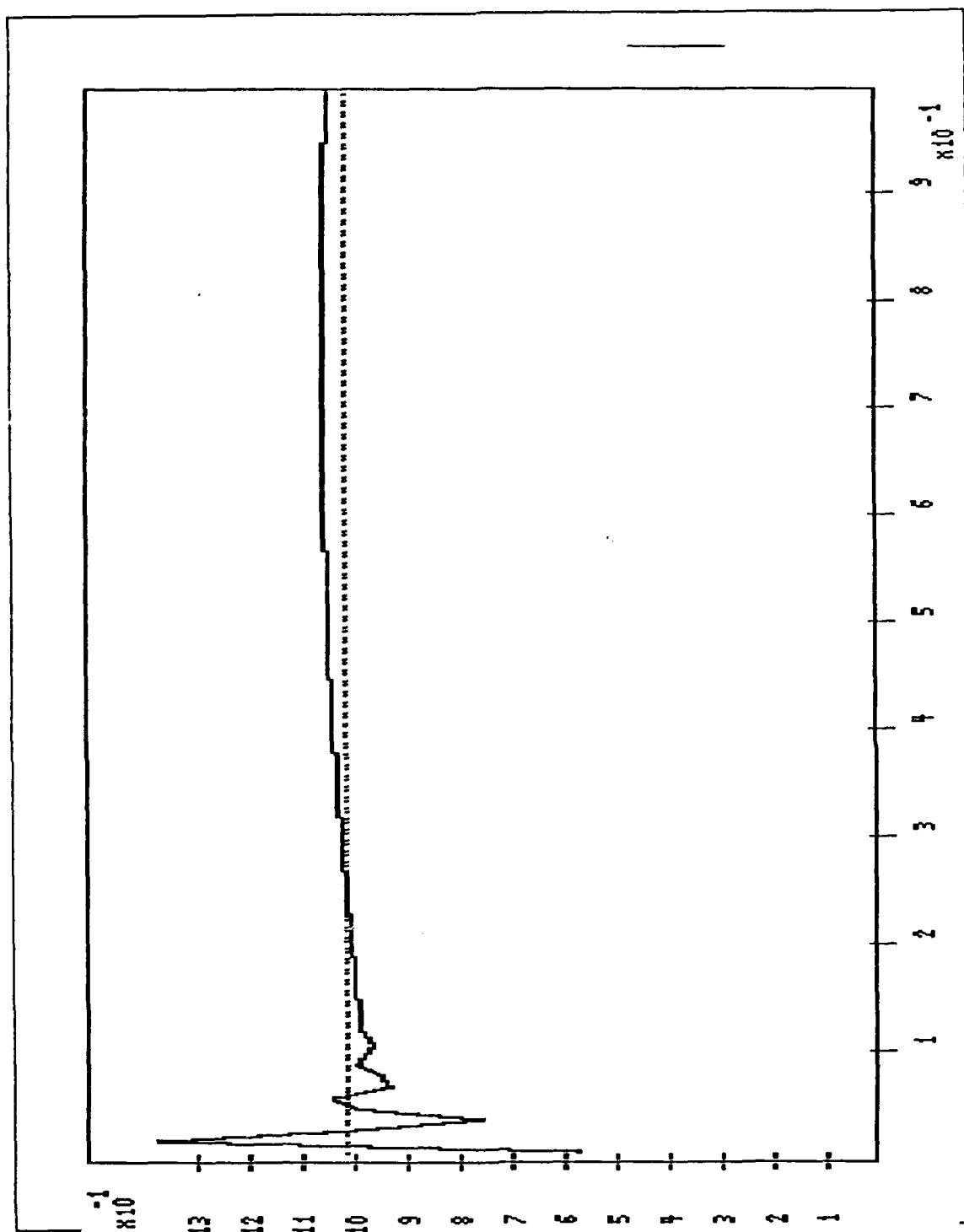


Figure 104. ALCON Simulation of Second Problem.

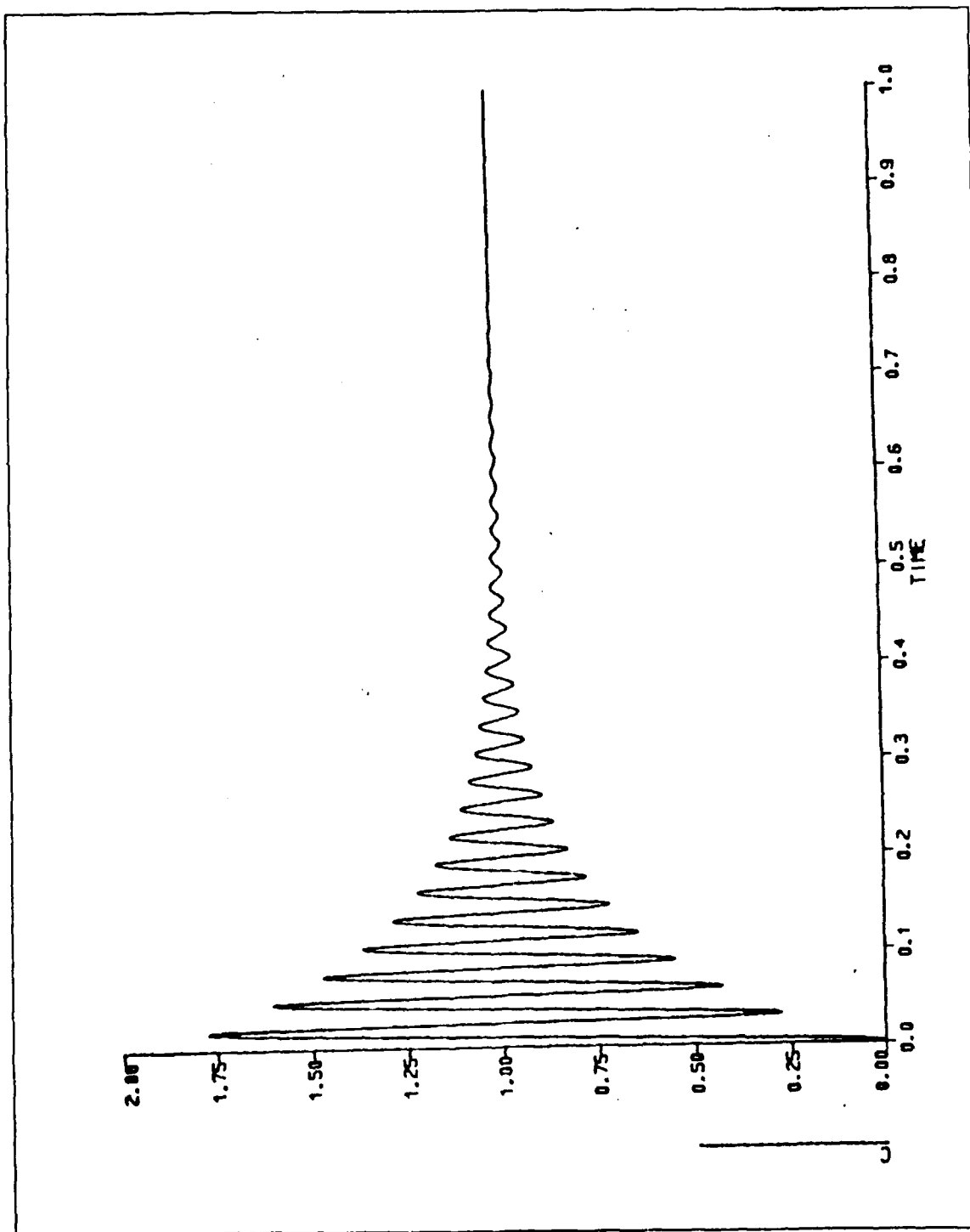


Figure 105. Second DSL Simulation of First Problem

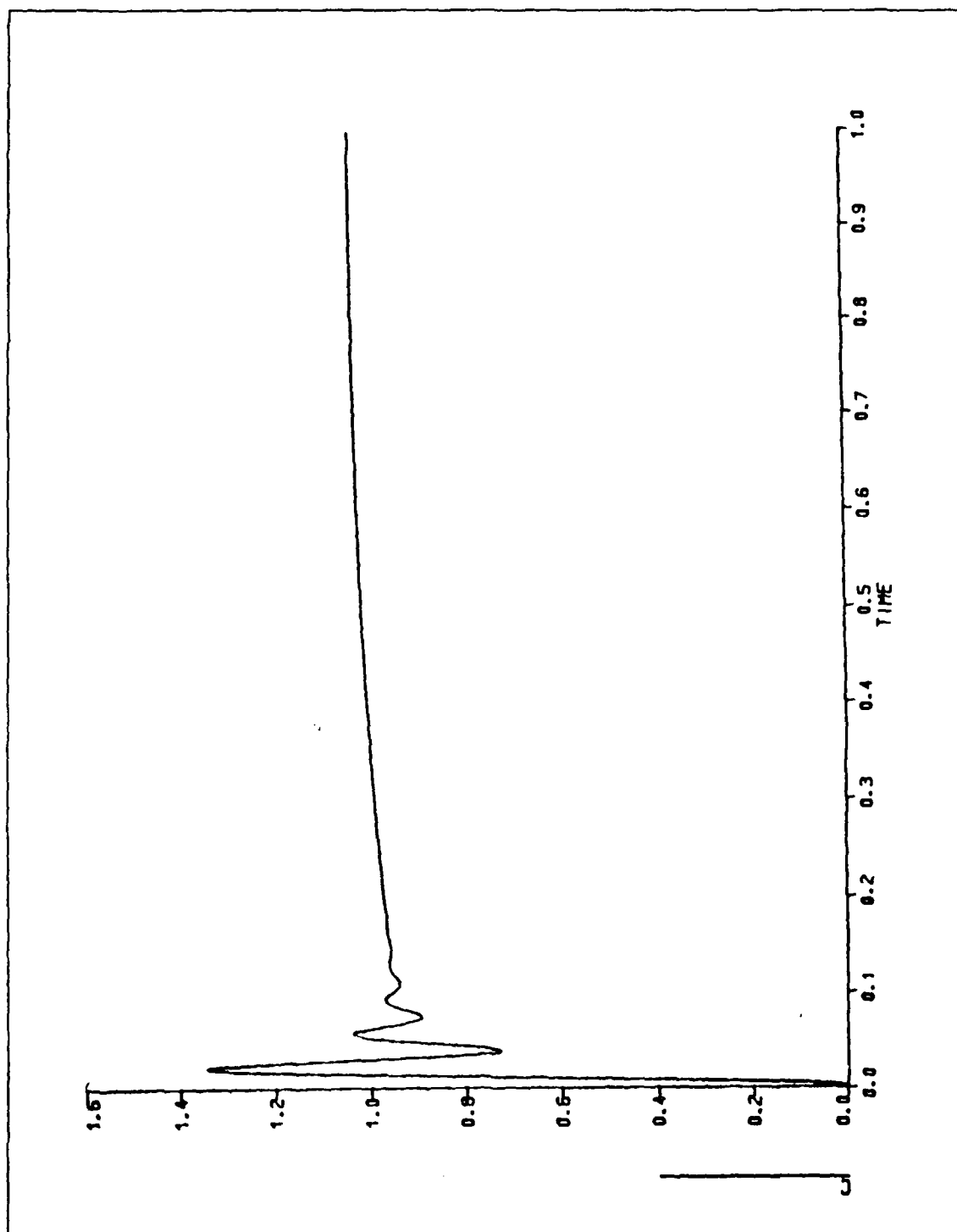


Figure 106. Second DSL Simulation of Second Problem

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

In this thesis, the development of a procedure to design the feedback compensators by using Root-Locus methods and pole placement methods has been presented. Simulation results and worked out examples have shown that except for extremely high order systems designing the compensators with the Root-Locus method can be used for any plant. The most important result is to design them using pole placement concepts and to be able to use them either in cascade or in the forward path.

Simulation results have also shown that the compensator can be determined by a Function Minimization subroutine directly from the transfer function of an ideal plant based on given specifications.

B. RECOMMENDATIONS

1. Further research should investigate methods to satisfy certain specifications while designing compensators with the Root-Locus method.
2. In this thesis, it has been shown that Function Minimization can be used to design compensators, but the user should write his own program according to the parameters. A package program may be written for the main frame to make the procedure interactive and faster.
3. Further research may investigate the effect of gain variations on the root locations while using a pole placement method so one can use directly multiplication of desired roots as $H(s)$.

APPENDIX A. CONSTRAINT PARAMETERS OF FUNCTION MINIMIZATION

Subroutine HOOKE uses the Hooke and Jeeves "direct search" method to find the local maximum or minimum of a multi-parameter criterion function, CF. [Ref. 3].

The algorithm evaluates CF at a base point, $X = (X(1), \dots, X(N))$, then perturbs each parameter in turn, by the amount $\pm STEP(I)$, and evaluates CF at each new point.

If none of these points produces a better value of CF, then the stepsizes are decreased by a factor of BETA, and the process is repeated.

To use HOOKE, the user must initialize the following arguments and, in the main program, CALL HOOKE(X,STEP,N,ITMAX,CFTOL,ALPHA,BETA,CF,Q,QQ,W,IPRINT,MINMAX)

All of these arguments must be initialized in MAIN, except for X, CF, Q, QQ, and W. Recommended values for ALPHA and BETA are ALPHA = 2., BETA = 0.5. All of the arrays, i.e., X, STEP, Q, QQ and W, must be declared and dimensioned in MAIN.

The arguments and their meanings may be explained as following.

X = The array of N parameter values. The user must supply the initial guesses, either in the DSL program or in MAIN.

STEP = An array of dimension N containing the initial step sizes to be used in the search.

N = The number of parameters (a positive integer, at most 15).

ITMAX = The maximum number of function calls to be performed.

CFTOL = The error in the criterion function to be reached before the program terminates (difference between the current value and the previous stage value).

ALPHA = The factor of $(Y - X)$ which is added to Y to get XNEW; a number greater than or equal to 1.

BETA = The stepsize reduction factor; a number between 0 and 1.

CF = The value of the criterion function.

Q, QQ, W = Arrays of dimension N, to be used as work space. They must be declared and dimensioned in the main program.

IPRINT = An integer flag. 0 for no intermediate printout, 1 for intermediate printout of X, CF, the number of function evaluation and notification of step-reduction.

MINMAX = An integer flag, -1 searches for a minimum, +1 searches for maximum.

APPENDIX B. COMPUTER PROGRAM FOR FUNCTION MINIMIZATION

*This program finds the necessary parameters to fit the function to some
*special transient response. By changing zeta and omgn the standart time
*response may be generated.

D COMMON /HANDJ/FLAG,ERRFN,A1,A2,B1,B2

TITLE Rocket parameters

* FUNCTION MINIMIZATION

*Put initial conditions according to the previous assuming parameters.

INCON A10=2. ,A20=5. ,B10=1. ,B20=100.

*Define your arrays to simulate your system

ARRAY P(1),Q(2),A(1),B(2),D(5),E(5),F(1),G(3),H(9),L(11),K(2),...
M(2),Y(3),Z(3),S(2),T(4)

*Put the values for the array coefficients

TABLE A(1)=333. ,B(1-2)=1. ,333. ,D(1-5)=1. ,92. ,18340. ,984000. ,...
51200000. ,E(1-5)=1. ,60. ,19510. ,558300. ,74368000. ,...
F(1)=2750. ,G(1-3)=1. ,42. 2,2750. ,H(1-9)=1. ,40. 652,...
2949. 22,1048754. 7,1. 86426E+08,3. 9622E+09,1. 17521E+11,...
2. 12662E+12,1. 10017E+14,L(1-11)=1. ,53. 12,30506. 757,...
1378867. ,1. 83946E+08,5. 2509E+09,3. 42441E+11,2. 85168E+12,...
1. 43994E+14,1. 06498E+13,-3. 42095E+14,K(1-2)=13. 3,39. 9,...
M(1-2)=1. ,30. ,P(1)=25. ,Q(1-2)=1. ,100. ,Y(1)=1. ,Z(1)=1. ,...
S(1-2)=10. ,50. ,T(1-4)=1. ,4. ,8. ,0.

PARAM R=1.

*Changing zeta and omgn changes the specifications of your system

CONST ZETA=0. 40, OMGN=6. 0,IC=0. 0

INITIAL

*This part of the program generates the standart time response

TOTERR=0. 0

C1=SQRT(1. 0-ZETA*ZETA)

C2=OMGN*ZETA

C3=OMGN*C1

PHI=ATAN(C1/ZETA)

* ASSIGN INITIAL GUESS FOR FUNCTION MINIMIZATION

*This part gives the starting point to the program

IF (FLAG. GE. 0.) GO TO 10

A1=A10

IF (FLAG. GE. 0.) GO TO 10

A2=A20

IF (FLAG. GE. 0.) GO TO 10

B1=B10

IF (FLAG. GE. 0.) GO TO 10

B2=B20

10 CONTINUE

*Put your variable coefficients here so program can change it

Y(2)=A1

Y(3)=A2

Z(2)=B1

Z(3)=B2

*Write normal simulation program for your system

DERIVATIVE

```

X0=TRNFR(8,10,0.,H,L,X2*2.849)
X2=TRNFR(0,2,0.,F,G,X3)
X3=TRNFR(4,4,0.,D,E,X4)
X4=TRNFR(0,1,0.,A,B,X6)
X6=15.*ERG
ERG=ER-X9
X9=TRNFR(2,2,0.,Y,Z,X0)
ER=R-X0
DYNAMICS
*Standart time response is generated by this formula
STD=1.0-(EXP(-C2*TIME)/C1)*SIN(C3*TIME+PHI)
ERR=X0-STD
ERRSQ=ERR*ERR
TOTERR=INTGRL(0.,ERRSQ)
*As a cost function integral of error square is used
TERMINAL
*This part finishes the calculation when min value is reached.
ERRFN = TOTERR
CONTRL FINTIM=2.,DELT=0.01
END
STOP
FORTRAN
*From now on just declare your step values and amount of variables
*   MAIN PROGRAM FOR FUNCTION MINIMIZATION
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(4),STEP(4),Q(4),QQ(4),W(4)
C   X(1)=K,X(2)=P INITIAL VALUES IN DSL ROUTINE
      STEP(1)=0.5D0
      STEP(2)=5.0D0
      STEP(3)=0.5D0
      STEP(4)=10.0D0
      N=4
      ITMAX = 100
      CFTOL = 1.0D-6
      ALPHA = 2.0D0
      BETA = 0.5D0
      IPRINT = 0
      MINMAX = -1
      CALL HOOKE(X,STEP,N,ITMAX,CFTOL,ALPHA,BETA,
*           CF,Q,QQ,W,IPRINT,MINMAX)
      STOP
      END

```

LIST OF REFERENCES

1. Han, Kuang-Wei, *Model Reduction for Keeping Gain Margin and Phase Margin Unchanged*. IEEE Transactions on Industry Applications, Vol. IA-22, Number 1, pp. 51 - 57, 1986.
2. Unlu, Ismail, *State Variable Systems*, MSEE Thesis, Naval Post-graduate School, Monterey, CA, December, 1987.
3. I.B.M., *Dynamic Simulation Language*, Chapter 5, pp. 56 - 61, 1987.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Chairman, Code 62 Naval Postgraduate School Monterey, CA 93943-5000	1
4. Professor George J. Thaler, Code 62Tr Naval Postgraduate School Monterey, CA 93943-5000	1
5. Professor Hal A. Titus, Code 62Ts Naval Postgraduate School Monterey, CA 93943-5000	1
6. Levent Korkmaz Alidede Sok. 29, 13 Asagi Ayranci Ankara, TURKEY	2
7. Deniz Kuvvetleri Komutanligi Personel Egitim Daire Baskanligi Bakanliklar - Ankara, TURKEY	3
8. Deniz Harp Okulu Ogretim Kurulu Baskanligi Tuzla - Istanbul, TURKEY	2
9. O. D. T. U. Kutuphanesi Orta Dogu Teknik Universitesi Ankara, TURKEY	1
10. Hacettepe Universitesi Kutuphanesi Hacettepe Universitesi, Beytepe Ankara, TURKEY	1
11. Naval Ocean Systems Center Attn. Library San Diego, California 92152	1